

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені ІГОРЯ СІКОРСЬКОГО»

Теплоенергетичний факультет

Кафедра автоматизації проектування енергетичних процесів і систем

До захисту допущено

Завідувач кафедри

О.В. Коваль

(підпис)

(ініціали, прізвище)

“ ” 2020р.

**ДИПЛОМНА РОБОТА**

на здобуття ступеня бакалавра

з напрямку підготовки 121 Інженерія програмного забезпечення

на тему “Система ранжирування територій України за рівнем соціально-екологічних ризиків людського розвитку”

Виконав (-ла): студент (-ка) 4 курсу, групи ТВ-зб1

Агафонова Софія Валеріївна

(прізвище, ім'я, по батькові)

(підпис)

Керівник доц., к.е.н. Караєва Наталія Веніамінівна

(посада, вчене звання, науковий ступінь, прізвище та ініціали)

(підпис)

Консультант

(назва розділу)

(вчені ступінь та звання, прізвище, ініціали)

(підпис)

Рецензент

(посада, вчене звання, науковий ступінь, прізвище та ініціали)

(підпис)

Засвідчую, що у цій дипломній роботі немає  
запозичень з праць інших авторів без  
відповідних посилань.

Студент

(підпис)

Київ – 2020 року

**Національний технічний університет України  
“Київський політехнічний інститут імені Ігоря Сікорського”**

Факультет теплоенергетичний

Кафедра автоматизації проектування енергетичних процесів і систем

Рівень вищої освіти перший рівень

Напрямок підготовки: 121 Інженерія програмного забезпечення

Спеціалізація: Програмне забезпечення розподілених систем

ЗАТВЕРДЖУЮ

Завідувач кафедри

\_\_\_\_\_ О.В. Коваль  
(підпис)

” \_\_\_\_ ” \_\_\_\_\_ 2020р.

**ЗАВДАННЯ**

**на дипломну роботу студенту**

Агафоновій Софії Валеріївні

(прізвище, ім'я, по батькові)

1. Тема роботи “Система ранжирування територій України за рівнем соціально-екологічних ризиків людського розвитку”

керівник роботи Караєва Наталія Веніамінівна, доц., к.е.н.

(прізвище, ім'я, по батькові науковий ступінь, вчене звання)

затверджена наказом вищого навчального закладу від ” \_\_\_\_ ” \_\_\_\_ 2020 р. № \_\_\_\_

2. Строк подання студентом роботи \_\_\_\_\_

3. Вихідні дані до роботи: мова програмування Python та фреймворк Django \_\_\_\_\_

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) Проаналізувати існуючі методи вимірювання рівня людського розвитку і відповідні програмні рішення ранжирування територій, спроектувати структуру бази даних, спроектувати архітектуру програми, розробити програмне забезпечення, реалізувати інтерфейс користувача системи.

5. Перелік ілюстративного матеріалу

Актуальність, мета розробки, вхідні та вихідні дані, постановка задачі, ризик-чинники сталого людського розвитку України, діаграма прецедентів, засоби реалізації, архітектура системи, концептуальна схема БД, інтерактивна карта індексу людського розвитку, результати побудови діаграм чинників людського розвитку, інтерфейс адміністрування табличних даних, висновки.

## 6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

## 7. Дата видачі завдання "11" жовтня 2019 р.

### КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів роботи	Примітки
1.	Затвердження теми роботи	19.12.2019 р.	
2.	Вивчення та аналіз задачі	01.03.2020 р. – 20.03.2020 р.	
3.	Розробка архітектури та загальної структури системи	20.03.2020 р. – 25.04.2020 р.	
4.	Розробка структур окремих підсистем	25.04.2020 р. – 10.05.2020 р.	
5.	Програмна реалізація системи	10.05.2020 р. – 25.05.2020 р.	
6.	Оформлення пояснювальної записки	25.05.2020 р. – 10.06.2020 р.	
7.	Захист програмного продукту	10.06.2020 р.	
8.	Передзахист	10.06.2020 р.	
9.	Захист	17.06.2020 р.	

Студент

\_\_\_\_\_

(підпис)

Агафонова С.В.

(прізвище та ініціали,)

Керівник роботи

\_\_\_\_\_

(підпис)

Караєва Н.В.

(прізвище та ініціали,)

## **ВІДГУК**

### **керівника дипломної роботи**

освітньо-кваліфікаційного рівня „бакалавр”

виконаної на тему: “Система ранжирування територій України за рівнем соціально-екологічних ризиків людського розвитку”

студенткою Агафоновою Софією Валеріївною

Дипломна робота Агафонової С. В. присвячена розробці системи ранжирування територій України за рівнем соціально-екологічних ризиків людського розвитку. Тема дипломної має практичну спрямованість у сфері аналізу екологічних та соціальних ризиків і є актуальною.

У ході виконання роботи студенткою було проведено аналіз предметної області та визначено завдання, які мають вирішуватись розробленою системою. Студентка проявила високу обізнаність у прийнятті сучасних рішень, уміння аналізувати статистичну інформацію, приймати правильні технологічні рішення.

Студентка показала досить високий рівень застосування на практиці набутих знань і навиків, показала високий рівень підготовки.

Дипломна робота виконана на високому рівні, пояснювальна записка оформлена належним чином у відповідності до прийнятих стандартів і студентка Агафнова С. В. заслуговує на присвоєння освітнього-кваліфікаційного рівня бакалавр з інженерії програмного забезпечення зі спеціальності 121 – “Інженерія програмного забезпечення”, за спеціалізацією “Програмне забезпечення розподілених систем”.

Керівник дипломної роботи

доц., к.е.н.

(посада, вчені звання, ступінь)

\_\_\_\_\_  
(підпис)

Карасєва Н.В.

(ініціали, прізвище)

## РЕЦЕНЗІЯ

### на дипломну роботу

освітньо-кваліфікаційного рівня „бакалавр”

виконаної на тему: “Система ранжирування територій України за рівнем соціально-екологічних ризиків людського розвитку”

студенткою Агафоновою Софією Валеріївною

Актуальність представленої на рецензію дипломної роботи Агафонові С.В. “Система ранжирування територій України за рівнем соціально-екологічних ризиків людського розвитку” має практичну спрямованість у сфері аналізу екологічних та соціальних ризиків і є актуальною.

Пояснювальна записка містить опис результатів виконання дипломної роботи, відповідає затвердженій темі та виконана відповідно до завдання.

Робота містить теоретичне обґрунтування предметної області, опис структури програмної системи, обґрунтування вибору програмних та технічних засобів реалізації, а також методику роботи користувача з програмною системою.

Пояснювальна записка містить опис архітектури програмних модулів, приклади заповнення бази даних, екранні форми, приклади діаграм, що побудовані програмною системою.

Агафнова С.В. у своїй роботі використовувала мову програмування Python, фреймворк Django та бібліотеки FusionCharts та Bootstrap.

Матеріал у бакалаврській роботі викладено досить чітко та ясно. Дипломна робота оформлена якісно та повністю відповідає вимогам, які висувають до дипломних робіт бакалаврів.

Програмна реалізація виконана у повному обсязі та заслуговує оцінки «відмінно», а студентка Агафнова С.В. заслуговує на присвоєння освітнього-кваліфікаційного рівня бакалавр з інженерії програмного забезпечення зі спеціальності 121 – “Інженерія програмного забезпечення”, за спеціалізацією “Програмне забезпечення розподілених систем”.

Рецензент

---

(посада, вчені звання, ступінь)

---

(підпис)

---

(ініціали, прізвище)

## **АНОТАЦІЯ**

Метою дипломної роботи є розробка системи ранжирування територій України за рівнем соціально-екологічних ризиків людського розвитку. Система дозволяє на основі даних державної статистики будувати інтерактивні карти та діаграми чинників людського розвитку. Це надає можливість порівнювати рівень людського розвитку та його окремих чинників у кожному регіоні, тому може слугувати підґрунтям для аналізу та прогнозування соціально-екологічних ризиків людського розвитку України.

Записка складається з 5 розділів, 57 сторінок, 39 рисунків, 4 таблиці та 29 використаних джерел.

## **ABSTRACT**

The purpose of the thesis is to develop a system of ranking the territories of Ukraine according to the level of environmental and social risks of human development. The system allows us to build interactive maps and diagrams of human development factors based on official statistics. It provides an opportunity to compare the level of human development and human development factors in each region, so it can serve as a basis for analysis and forecasting of environmental and social risks of human development in Ukraine.

The note consists of 5 sections, 57 pages, 39 figures, 4 tables and 29 sources used.

# ЗМІСТ

Перелік умовних позначень, скорочень і термінів .....	8
Вступ.....	10
1. Постановка задачі створення системи ранжирування територій України за рівнем соціально-екологічних ризиків людського розвитку .....	12
2. Огляд існуючих методів вимірювання рівня людського розвитку і відповідних програмних рішень ранжирування територій .....	14
2.1. Передумови і ризик-чинники сталого людського розвитку .....	14
2.2. Методична основа вимірювання рівня людського розвитку .....	18
2.3. Веб - застосунок United Nations Development Programme.....	20
3. Засоби розробки.....	23
3.1. Обґрунтування вибору веб-застосунку як типу програмного забезпечення.....	23
3.2. Вибір середовища розробки .....	23
3.3. Вибір мови програмування та бібліотек .....	24
3.4. Обґрунтування вибору бази даних .....	28
4. Опис програмної реалізації .....	34
4.1. Архітектура програмної системи.....	34
4.2. Опис бази даних .....	36
5. Методика роботи користувача з програмною системою .....	44
5.1. Системні вимоги та інсталяція програмного продукту.....	44
5.2. Сценарії роботи користувача з системою.....	44
Висновки .....	60
Список використаних джерел .....	61
Додаток 1 .....	64
Додаток 2 .....	71
Додаток 3 .....	79

# ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ІЛР – Індекс Людського Розвитку

БД – База Даних

ПЗ – Програмне Забезпечення

SQL, Structured Query Language – декларативна мова програмування, застосовується для створення, модифікації та управління даними в реляційній базі даних, керованій відповідною системою управління базами даних.

NoSQL, Structured Query Language – термін, що позначає ряд підходів, спрямованих на реалізацію систем управління базами даних, що мають суттєві відмінності від моделей, що використовуються в традиційних реляційних СУБД з доступом до даних засобами мови SQL.

СУБД – Система Управління Базами Даних

ORM, Object-Relational Mapping – технологія програмування, яка зв'язує бази даних з концепціями об'єктно-орієнтованих мов програмування, створюючи "віртуальну об'єктну базу даних".

Back end – частина комп'ютерної системи або програми, до якої безпосередньо не звертається користувач. Зазвичай відповідає за зберігання та маніпулювання даними.

Front end – інтерфейс для взаємодії між користувачем і back end.

JSON, JavaScript Object Notation – формат текстового обміну даними, зручний для читання, використовується для представлення структур даних (масив, словник тощо) у текстовому вигляді

XML, eXtensible Markup Language – розширювана мова розмітки.

RSS, Rich Site Summary – один з підмножину XML-форматів, призначених для опису стрічок новин, анонсів статей, змін в блогах тощо.

SVG, Scalable Vector Graphics – мова розмітки масштабованої векторної графіки, що входить в підмножину розширюваної мови розмітки XML, призначений для опису двовимірної векторної і змішаної векторно-растрової графіки в форматі



XML.

HTML, HyperText Markup Language – стандартизована мова розмітки документів.

CSS, Cascading Style Sheets – формальна мова опису зовнішнього вигляду документа, написаного з використанням мови розмітки.

DOM, Document Object Model — це незалежний від платформи і мови програмний інтерфейс, що дозволяє програмам і скриптам отримати доступ до вмісту HTML-, XHTML- і XML-документів, а також змінювати зміст, структуру і оформлення таких документів.

API, Application Programming Interface — прикладний програмний інтерфейс.

## ВСТУП

В 1992 році на Конференції Організації Об'єднаних Націй з навколишнього середовища і розвитку (Саміт "Планета Земля") в Ріо-де-Жанейро урядами 179 країн (у тому числі Україною) було прийнято офіційну стратегію переходу до сталого розвитку як "Порядок денний на XXI століття" [1]. Концепція людського розвитку є домінантою сучасної концепції сталого розвитку. Саме соціально-екологічна криза, що загострилася наприкінці 1970-х – початку 1980-х років XX століття, стало підґрунтям формування уявлень про сталий розвиток як процес реалізація прав та потреб людей, розвиток та реалізація їхніх здібностей і талантів. Якщо нерівність у людському розвитку подалі зростатиме, прагнення Порядку денного в сфері сталого розвитку до 2030 року не будуть втілені.

Для забезпечення умов людського розвитку необхідно формувати та впроваджувати соціально-екологічну модель сталого розвитку на засадах соціального благополуччя, економічного зростання та екологічної безпеки [2].

В якості інструменту вимірювання та ранжирування країн за рівнем соціально-економічних і екологічних досягнень на початку 1990-х років експертами Програми розвитку Організації Об'єднаних Націй (ПРООН) було запропоновано розраховувати індекс людського розвитку (ІЛР) (Human Development Index – HDI) [3]. Індекс людського розвитку – це інтегральний показник тривалості життя, якості освіти та показників доходу на душу населення. Починаючи з 1990 року ПРООН випустила понад 800 глобальних, регіональних, національних і субнаціональних доповідей про людський розвиток, в яких представлено результати вимірювання ІЛР, моніторингу його динаміки та наведено шляхи мінімізації соціально-економічних і екологічних ризиків сталого людського розвитку. Розрахунки ІЛР для України були вперше включені у Звіт ПРООН з людського розвитку у 1993 році [4].

Диференціація регіонів України за рівнем людського розвитку – найбільш комплексний індикатор оцінки ефективності регіональної політики [5]. Регіональна ефективна політика забезпечення прийнятного рівня людського розвитку, в значній мірі, залежить від рівня інформаційно-аналітичного забезпечення. Тому актуальність

даної роботи полягає у тому, що розробка системи ранжування та графічного відображення числових показників людського розвитку допомагає прослідкувати за динамікою змін всіх аспектів людського розвитку по кожному регіону окремо впродовж певного періоду, а також по всім регіонам країни за обраний рік. Система надає можливість порівняти рівень людського розвитку у кожному регіоні, тому може слугувати підґрунтям для аналізу та прогнозування соціально-екологічних ризиків людського розвитку.

Існуючі закордонні програмні продукти не адаптовані під нормативно-правові вимоги українського законодавства і надають лише аналіз показників по країні в цілому, а не по регіонах. Українські аналоги надають лише стислі статистичні дані, без графічної візуалізації.

Робота містить 5 розділів.

У першому розділі описана мета виконання дипломної роботи, призначення веб-застосунку та задачі, які він виконує.

У другому розділі оглядається існуюче програмне забезпечення для аналізу ризиків людського розвитку та побудови діаграм на їхній основі.

У третьому розділі описуються обрані засоби розробки та обґрунтовується їхнє застосування у розробці веб-застосунку.

Четвертий розділ містить опис програмної реалізації задачі, архітектури розробленої системи, схему та опис бази даних.

У п'ятому розділі оглядається взаємодія користувача з системою.

У висновках описані результати розробленого веб-застосунку та навички і знання, що були здобуті під час виконання роботи.

# **1. ПОСТАНОВКА ЗАДАЧІ СТВОРЕННЯ СИСТЕМИ РАНЖИРУВАННЯ ТЕРИТОРІЙ УКРАЇНИ ЗА РІВНЕМ СОЦІАЛЬНО-ЕКОЛОГІЧНИХ РИЗИКІВ ЛЮДСЬКОГО РОЗВИТКУ**

Концепція розвитку людського потенціалу є одним з найбільш відомих інтелектуальних продуктів, розроблених ПРООН. Дана система поглядів орієнтована на підвищення якості життя людини, розширення і вдосконалення її можливостей у всіх областях.

Ризик-чинники, що впливають на рівень людського розвитку різняться для кожної країни, тому виникає потреба у розробці веб-застосунку, що би допомагав у побудові діаграм на основі числових значень ІЛР для України, що б допомагало прослідкувати зміни показників по регіонах та подальший аналіз ризиків людського розвитку.

Програмне забезпечення має вирішувати наступні задачі:

- надання користувачеві теоретичних відомостей про аспекти людського розвитку України;
- відображення на карті значень інтегрального індексу кожного регіону за обраний рік;
- побудова графіків по кожному показнику людського розвитку України
- збереження побудованих графіків;

Вхідними даними для системи є статистичні дані показників людського розвитку.

Вихідними даними є побудовані графіки, що відображають зміну показників за певний період або за кожен рік окремо.

Користувачами програмного забезпечення можуть бути міжнародні та вітчизняні експертні організації, науково-дослідницькі інститути та вищі навчальні заклади.

На рисунку 1.1 зображено UML-діаграму прецедентів. Дана діаграма описує типові взаємодії між користувачами системи і з самою системою.

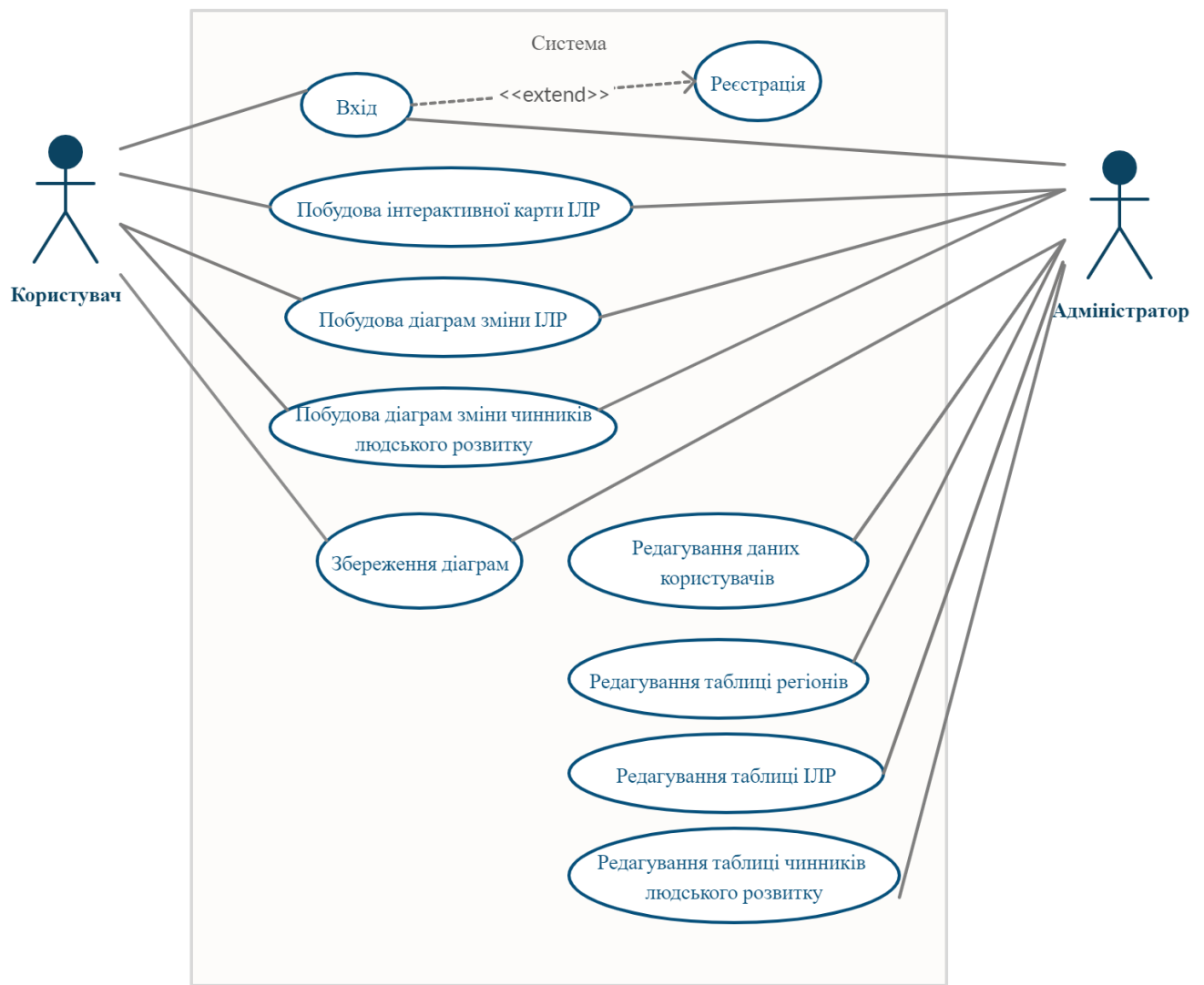


Рисунок 1.1 – UML-діаграма прецедентів

Для розробки програмного продукту буде використано середовище PyCharm, об'єктно-орієнтовану мову програмування Python на базі веб-фреймворку Django та реляційну базу даних – PostgreSQL. Для роботи з діаграмами буде використовуватись модуль, написаний мовою Python на базі фреймворку Fusioncharts для побудови графіків. Front end частина написана з використанням HTML, CSS та JavaScript.

## **2. ОГЛЯД ІСНУЮЧИХ МЕТОДІВ ВИМІРЮВАННЯ РІВНЯ ЛЮДСЬКОГО РОЗВИТКУ І ВІДПОВІДНИХ ПРОГРАМНИХ РІШЕНЬ РАНЖИРУВАННЯ ТЕРИТОРІЙ**

У розділі розглянуто передумови і ризик-чинники сталого людського розвитку України. Обґрунтовано методичну основу формування інформаційної системи індикаторів і індексів за якими проводиться моніторинг і ранжирування територій за рівнем соціально-екологічних ризиків людського розвитку. Наведено характеристику програмного засобу, що використовується у світовій практиці для вищезазначеного типу задач.

### **2.1. Передумови і ризик-чинники сталого людського розвитку**

Сталим є такий розвиток, що задовольняє потреби теперішнього покоління не наражаючи на ризик здатність майбутніх поколінь задовольняти свої потреби. Концептуальні положення сталого людського розвитку глобального світу сформульовані у низці міжнародних документів (програм) і розроблені під егідою ООН (рисунок 2.1).

У 2015 року в рамках 70-ї сесії Генеральної Асамблеї ООН відбувся Саміт ООН зі сталого розвитку та прийняття Порядку денного розвитку після 2015 року, на якому було прийнято стратегічну програму "Перетворення нашого світу: порядок денний у сфері сталого розвитку до 2030 року". В програмі було затверджено 17 Цілей Сталого Розвитку та 169 завдань, що спрямовані на боротьбу з бідністю, голодом, хворобами, неосвіченістю, забрудненням довкілля. Україна, як член ООН, приєдналася до глобального процесу забезпечення сталого розвитку і Уряд України представив Національну доповідь "Цілі сталого розвитку: Україна" [6], яка визначає базові індикатори сталого розвитку і їх порогові значення (2015-2030 рр.) (рисунок 2.2).



Основні цілі сталого розвитку, що сформульовані у низці міжнародних документів (програм) і розроблені під егідою ООН

Декларація тисячоліття ООН, ухвалена Резолюцією 55/2 Генеральної Асамблеї 8 вересня 2000 року

Програма дій з подальшого впровадження Порядку денного на XXI століття, ухвалена Резолюцією A/S-19/29 Генеральної Асамблеї ООН 28 червня 1997 року

Йоганнесбурзька декларація зі сталого розвитку, ухвалена 4 вересня 2002 року

План виконання рішень Всесвітнього саміту на вищому рівні зі сталого розвитку, ухвалений 4 вересня 2002 року

Резолюція A/RES/60/1, прийнята Генеральною Асамблеєю ООН 16 вересня 2005 року, «Підсумковий документ Всесвітнього саміту 2005 року»; — Резолюція A/RES/66/288, прийнята Генеральною Асамблеєю на Конференції ООН зі сталого розвитку «Ріо+20» 27 червня 2012 року, «Майбутнє, якого ми прагнемо»

Резолюція A/RES/70/1, прийнята Генеральною Асамблеєю ООН 25 вересня 2015 року, про підсумковий документ «Перетворення нашого світу: Порядок денний сталого розвитку 2030»

Рисунок 2.1 – Міжнародні документи (програми), в яких визначено цілі сталого людського розвитку



Рисунок 2.2 – Скріншот ПРООН, де наведено глобальні цілі сталого розвитку України

Концептуальні положення людського розвитку, як базової основи концепції сталого розвитку, було розроблено всесвітньо відомим лауреат Нобелівської премії Амартія Сеном. Згідно запропонованої автором концепції розширення можливостей вибору людини [7], надання людям більшої свободи вибору такого способу життя, який вони вважають належним є водночас головною метою та базовим механізмом сталого людського розвитку.

Людський розвиток – це безперервний процес розширення вільного вибору людиною можливостей, зокрема: прожити довге якісне життя, здобути освіту, мати рівний доступ до засобів існування, безпечне довкілля (рисунок 2.3).



Рисунок 2.3 – Концептуальні положення людського розвитку

Щорічні міжнародні й вітчизняні дослідження людського розвитку свідчать про існування в Україні низки соціально-екологічних ризик-чинників (загроз) сталого людського розвитку, зокрема:

- депопуляція;
- низька тривалість життя і висока смертність населення, особливо у працездатному віці;
- низький рівень здоров'я населення під впливом соціально-економічних і екологічних чинників;



- низький рівень оплати праці;
- трудова міграція, особливо молоді;
- надмірне майнове розшарування населення;
- низькі стандарти життя в цілому;
- забруднення довкілля тощо.

Серйозні проблеми у різних сферах життєдіяльності населення України спричинені дією численних екологічних, економічних, соціальних і технологічних загроз (рисунок 2.4) і є гальмом на шляху втілення визнаних світовою спільнотою принципів сталого людського розвитку.

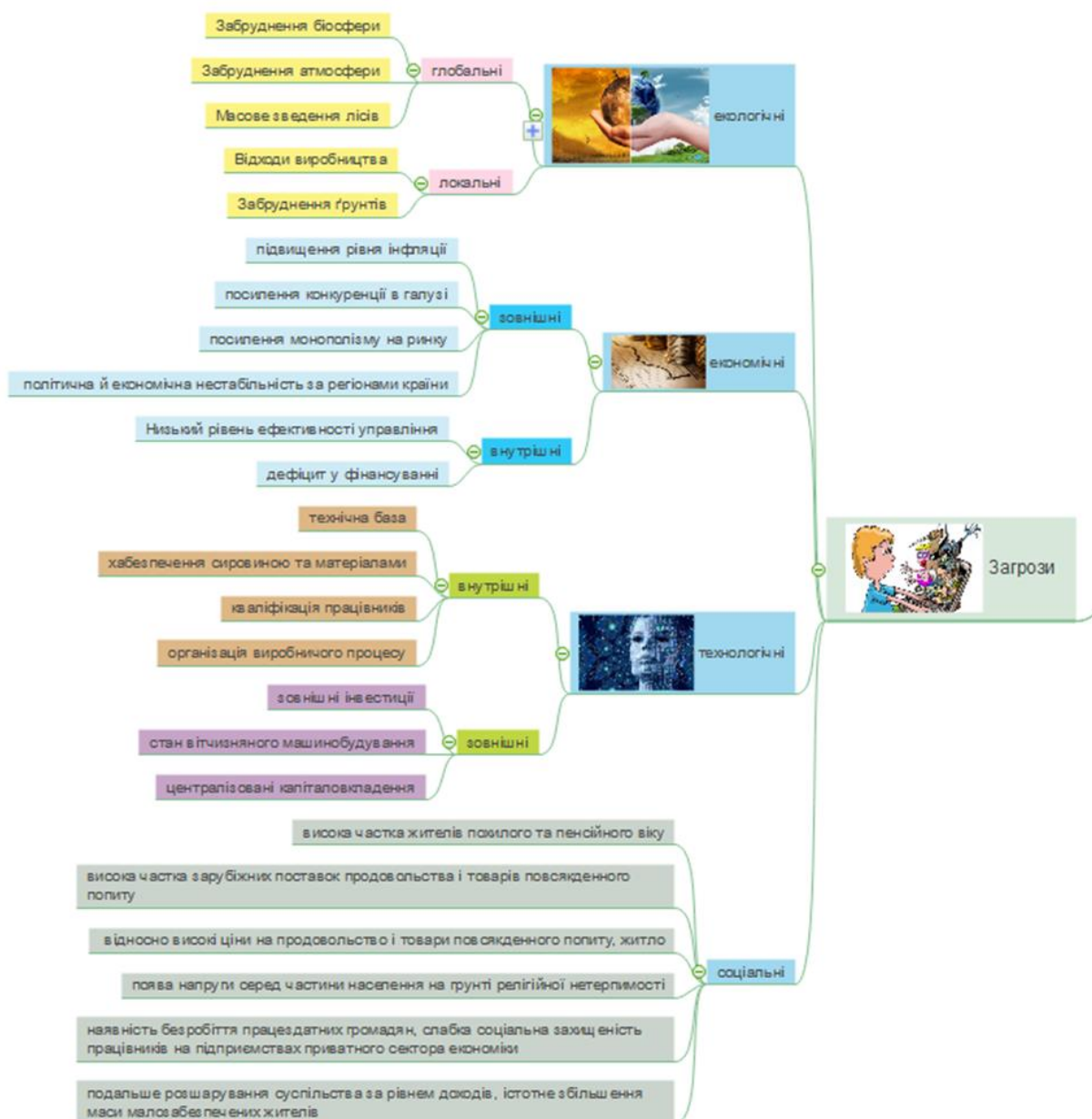


Рисунок 2.4 – Структура ризиків сталого людського розвитку

## 2.2. Методична основа вимірювання рівня людського розвитку

Для вимірювання людського розвитку, моніторингу його динаміки, ранжирування розробляється і використовується багатопараметричні системи відповідних статистичних показників (індикаторів) та інтегральних індексів. Індикатори людського розвитку представляють інформацію про зміни в екологічних, соціальних і економічних системах. Корисність індикатора залежить від специфіки контексту, в якому він використовується.

Розробка індикаторів людського розвитку здійснюється з використанням двох основних підходів:

- 1) побудова інтегрального індексу, на основі агрегованої оцінки стійкості. Агрегованість зазвичай здійснюється на основі трьох груп показників: еколого-економічних, еколого-соціально-економічних і власне екологічних. Кожен агрегований індекс обчислюється з використанням великої кількості індикаторів і наборів даних як кількісного, так і якісного характеру;
- 2) побудова системи часткових індикаторів стійкості. Формування індикаторів базується на побудові системи показників, що можуть відображувати окремі цілі людського розвитку.

На глобальному рівні складовими ІЛР є [8]:

- Індекс людського розвитку з урахуванням нерівності (IHDI);
- Індекс гендерного розвитку (GDI);
- гендерний характер Індексу нерівності (GII) (рисунок 2.5).

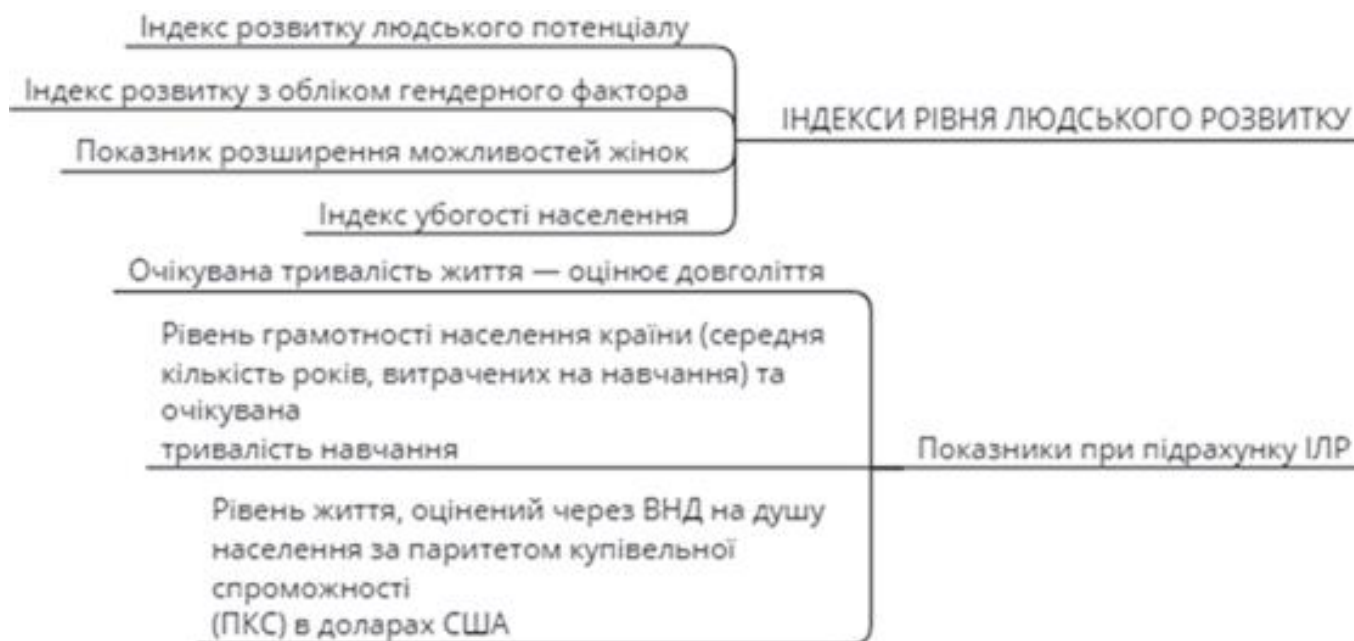


Рисунок 2.5 – Фрагмент структури індикаторів людського розвитку

У дослідженнях проблем сталого і людського розвитку також використовують класифікацію Світового банку, яка базується на головному економічному критерії – суспільній продуктивності праці.

За його схемою виокремлюються чотири типи країн [9]:

- 1) високорозвинуті країни (ВВП на душу населення понад 20 тис. дол.);
- 2) країни вище середнього рівня розвитку (10-20 тис. дол.);
- 3) країни нижче середнього рівня розвитку (5-10 тис. дол.);
- 4) бідні країни (менше 5 тис. дол.).

Крім глобального ІЛР, кожна країна розробляє свою національну систему індикаторів людського розвитку. Так, наприклад, в Україні науковцями Інституту демографії та соціальних досліджень ім. М.В. Птухи НАН України та фахівцями Державної служби статистики України здійснюється підготовка й видання щорічного статистичного бюлетеня "Регіональний людський розвиток" як Аналітична доповідь. В 2012 році рішенням Президії НАН України та колегії Державної служби статистики України від 13.06.2012 № 123-м затверджено нову Методику вимірювання регіонального людського розвитку [10]. До розрахунку регіонального ІЛР включено 33 показники, об'єднані у 6 блоків [11]:

- 1) відтворення населення;

- 2) соціальне становище;
- 3) комфортне життя;
- 4) добробут;
- 5) гідна праця;
- 6) освіта.

Також Методика передбачає розрахунок інтегрального показника стану навколишнього природного середовища. Метою розрахунку є оцінка стану довкілля (далі – інтегральний показник), що визначається за показниками стану атмосферного повітря, земельних та водних ресурсів через антропогенне навантаження і природний стан екосистем регіонів України.

Основними принципами відбору показників для виміру людського розвитку є [11]:

- придатність для щорічних розрахунків;
- забезпеченість наявною інформацією Державної служби статистики України;
- забезпечення надійності оцінок на регіональному рівні;
- відповідність специфіці проблем людського розвитку в Україні (диференціація окремих показників за віком, статтю та типом місцевості);
- однозначність трактування щодо впливу на людський розвиток;
- відсутність високої кореляції між окремими показниками.
- достатність статичної та динамічної варіації.

### **2.3. Веб - застосунок United Nations Development Programme**

United Nations Development Programme — це офіційний веб-ресурс, який створений ООН для оприлюднення офіційних даних, новин, таблиці індексу людського розвитку та інших статистичних даних. Однією з функціональних можливостей порталу є побудова графіків на основі даних індексу людського розвитку за певний період (рисунок 2.6) та за кожен рік окремо (рисунок 2.7).

## Human Development Data (1990-2018)

Select data by dimension, indicator, year and/or country to see a dynamic interactive visualization of the data (represented as line for trends, or bar for single years)

Dimension:

-Select-

Go to Indicator Page

Line

Bar

Clear All

Show All

Afghanistan  
Albania  
Algeria  
Andorra  
Angola  
Antigua and Barbuda  
Argentina  
Armenia  
Australia  
Austria  
Azerbaijan  
Bahamas  
Bahrain  
Bangladesh  
Barbados  
Belarus  
Belgium  
Belize  
Benin  
Bhutan  
Bolivia (Plurinational State of)  
Bosnia and Herzegovina  
Botswana  
Brazil  
Brunei Darussalam  
Bulgaria

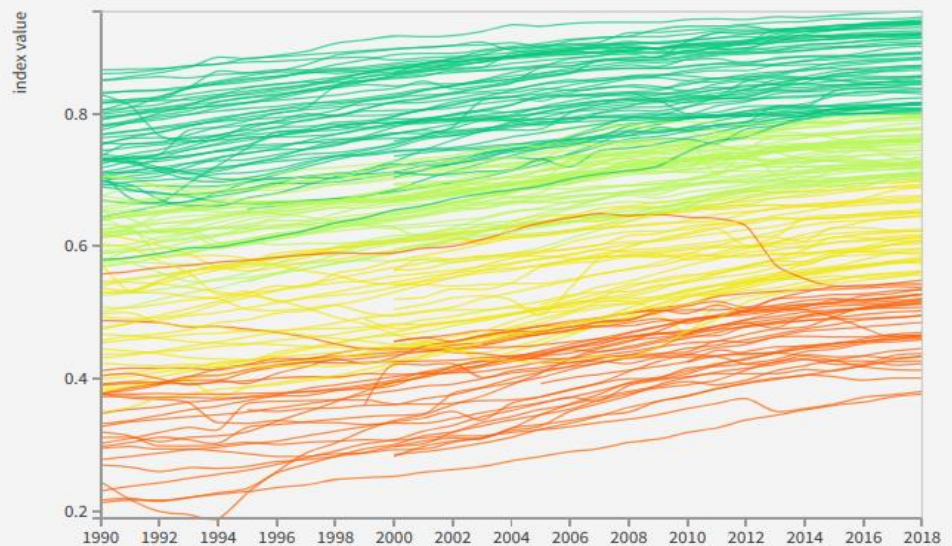


Рисунок 2.6 – Динаміка світового індексу людського розвитку

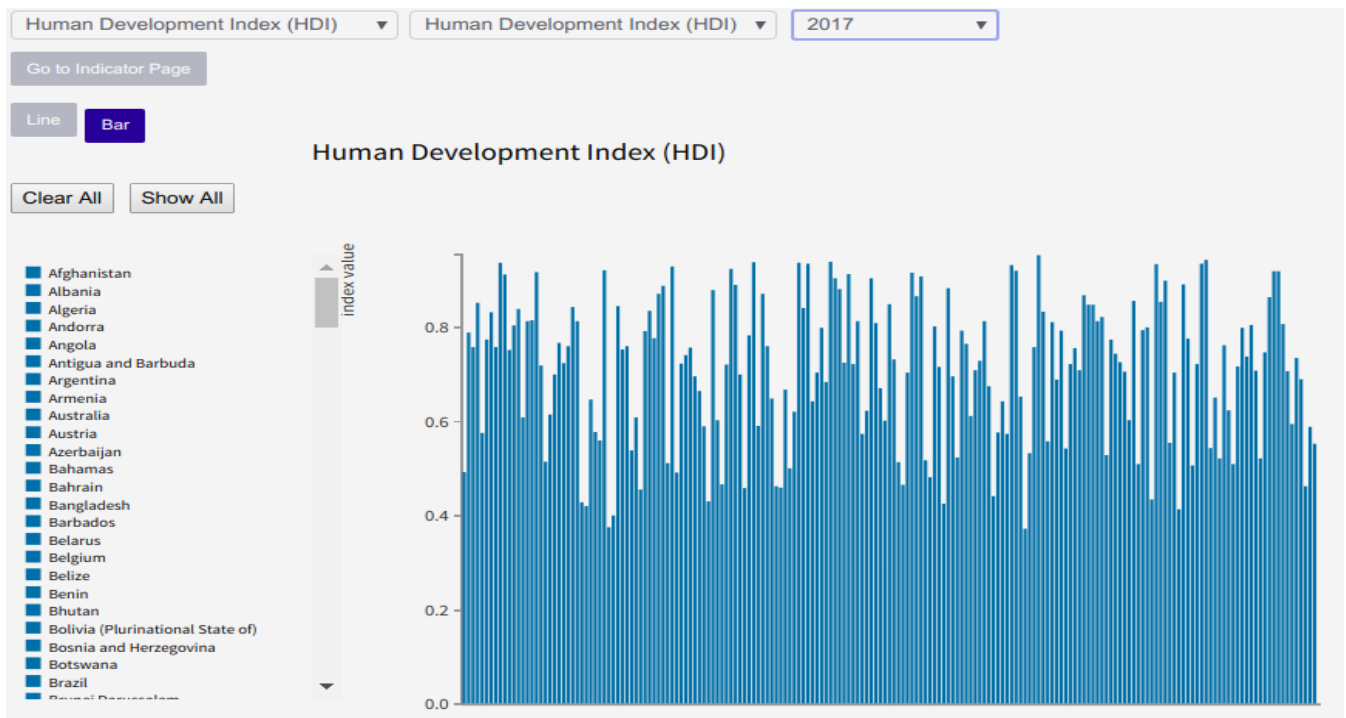


Рисунок 2.7 – Графік індексу людського розвитку за певний рік

Веб-застосунок має зручний інтерфейс, актуальні статистичні дані ООН, однак не може використовуватись для прогнозування ризиків людського розвитку регіонів України.

Дані числові показники індексу людського розвитку належать до країни в цілому, а тому не можуть слугувати підґрунтям до аналізу ризиків людського розвитку по регіонах України. Ще одним недоліком є те, що світові аспекти людського розвитку відрізняються від тих, що були прийняті законодавчими органами України.

Отже, проаналізувавши єдиний існуючий аналог розробляємої системи, можна стверджувати, що він в повній мірі не відповідає задачам розробляємої системи.

### **3. ЗАСОБИ РОЗРОБКИ**

Одним із найважливіших факторів при розробці програмного продукту є вибір засобів програмної реалізації. Від цього вибору залежить вартість та час розробки продукту, швидкість роботи системи та можливість підтримки продукту.

Для розробки даної системи була використана мова програмування Python, Back end написаний із використанням фреймворку Django та бібліотеки для побудови графіків Fusioncharts, Front end використовує HTML, CSS, Bootstrap та JavaScript.

#### **3.1. Обґрунтування вибору веб-застосунку як типу програмного забезпечення**

Система розроблена як веб-застосунок. Для такого вибору є декілька причин: по-перше програмне забезпечення має бути у публічному доступі, а по-друге працювати не залежно від операційної системи або пристрою користувача.

#### **3.2. Вибір середовища розробки**

Програмний продукт розроблявся у PyCharm. PyCharm – інтегроване середовище розробки (Integrated development environment або англ. IDE) для професійного програмування мовою Python розроблене компанією JetBrains. PyCharm надає широкі можливості для розробки консольних додатків, додатків з графічним інтерфейсом, систем, що використовують машинне навчання, а також веб-застосунків. Окрім мови програмування Python IDE підтримує JavaScript, CoffeeScript, TypeScript, Cython, SQL, HTML/CSS, мови шаблонів, AngularJS, Node.js та інші технології.

Середовище PyCharm було обране для розробки, адже підтримує багато сучасних фреймворків для веб-розробки: Django, Flask, Pyramid тощо. Це надає можливість розгортати сервери прямо з IDE, використовувати швидку навігацію по

коду, автодоповнення коду та виправлення помилок. Дане середовище розробки також надає можливість управління базами даних, аналізує всі об'єкти у базі та відображає їх, згруповані в папки за схемами, надає інтерфейс користувача для додавання та редагування таблиць, стовпців, індексів, обмежень тощо. PyCharm підтримує різні стилі і технології програмування: функціональне, декларативне, об'єктно-орієнтоване та інші [12].

### **3.3. Вибір мови програмування та бібліотек**

Python – високорівнева мова програмування загального призначення, орієнтований на підвищення продуктивності розробника і читання коду. Синтаксис ядра Python мінімалістичний. У той же час стандартна бібліотека включає великий обсяг корисних функцій [13].

Python підтримує структурний, об'єктно-орієнтоване, функціональне, імперативне і аспектно-орієнтоване програмування. Основні архітектурні риси – динамічна типізація, автоматичне керування пам'яттю, повна інтроспекція, механізм обробки виключень, підтримка багатопоточних обчислень, високорівневі структури даних. Підтримується розбиття програм на модулі, які, в свою чергу, можуть об'єднуватися в пакети [14]. Мова програмування Python надає дуже широкі можливості для розробника [15]:

- робота з XML/ HTML файлами;
- робота з HTTP запитам;
- GUI (графічний інтерфейс);
- створення веб-сценаріїв;
- робота з FTP;
- робота з зображеннями, аудіо та відео файлами;
- робототехніка;
- програмування математичних та наукових обчислень.

Django – це веб-фреймворк високого рівня, написаний мовою Python, щ



дозволяє швидко створювати захищені та підтримувані веб-застосунки. Він безкоштовний, має відкритий код та з детальною документацією. Django можна використовувати для створення майже будь-якого типу веб-додатків - від систем управління контентом та вікі-сайтів до соціальних мереж та новинних сайтів. Він може працювати з будь-яким середовищем клієнта і може доставляти вміст майже в будь-якому форматі (включаючи HTML, RSS-канали, JSON, XML тощо).

Django забезпечує безпечний спосіб керування обліковими записами та паролями користувачів, уникаючи поширених помилок, таких як розміщення інформації про сесію у файлах cookie там, де вона є вразливою (натомість файли cookie містять лише ключі, а фактичні дані зберігаються в базі даних) або безпосередньо зберігання паролів, замість хеша пароля. Django за замовчуванням забезпечує захист від багатьох вразливих місць, включаючи інжекцію SQL, міжсайтовий скриптинг, підробку міжсайтових запитів і клікджекінг.

Django використовує компонентну архітектуру “shared-nothing” (кожна її частина не залежить від інших, а тому може бути замінена або змінена). Чіткий розподіл між частинами означає, що Django може масштабуватись із збільшенням трафіку, додаючи апаратне забезпечення на будь-якому рівні: кешування серверів, серверів баз даних або серверів додатків.

Код Django написаний з використанням принципів дизайну та шаблонів, які заохочують створення підтримуваного та багаторазового використання коду. Зокрема, він використовує принцип “Do not Repeat Yourself” (DRY), тому немає зайвого дублювання, що зменшує кількість коду. Django також допомагає інтегрувати пов'язану функціональність у “багаторазові” програми та на нижчому рівні групувати пов'язаний код у модуль відповідно до шаблону контролера MVC (Model View Controller) [16].

FusionCharts – бібліотека графіків JavaScript, яка дозволяє створювати інтерактивні діаграми для веб-, мобільних та корпоративних додатків. FusionCharts Suite має 90+ типів діаграм та 1000+ карт для візуалізації даних.

FusionCharts допомагає створювати красиві інформаційні панелі для веб-проектів та мобільних пристроїв. Завдяки обширній документації, підтримці крос-

браузера та послідовному API додавати інтерактивні та чуйні діаграми простіше, ніж будь-коли. Від простих діаграм, таких як рядки та стовпці, до графіків, характерних для домену, таких як теплові карти, радіолокатори та графіки запасів [17].

Вибір даної бібліотеки обумовлений тим, що вона підтримує багато мов програмування і веб-фреймворків, в тому числі Django, має зручний API та проста у використанні.

Bootstrap – це фронт-фреймворк з відкритим кодом для швидшої та простішої веб-розробки. Bootstrap включає шаблони дизайну на основі HTML та CSS для типографії, форм, кнопок, таблиць, навігації, модалів, каруселей зображень та багатьох інших елементів, а також додаткові додатки JavaScript [18].

Причини використання Bootstrap:

- використання адаптивного дизайну, щоб дозволити веб-сайту пристосовуватися до різних розмірів екрана – мобільних, настільних та іншого;
- швидке та легке створення нових конструкцій;
- забезпечення сумісності з веб-браузерами.

Bootstrap можна звести до трьох основних файлів:

- bootstrap.css – рамка CSS;
- bootstrap.js – фреймворк JavaScript / jQuery;
- glyphicons – шрифт (набір шрифтів значків).

Крім того, Bootstrap вимагає jQuery для функціонування.

JavaScript була створена програмістом Brendan Eich з Netscape і представлена в грудні 1995 року під назвою LiveScript. Досить швидко мова була перейменована в JavaScript, хоча офіційною назвою JavaScript є ECMAScript. ECMAScript розробляється і підтримується Міжнародною організацією ЕСМА (Європейська асоціація виробників комп'ютерів) [19].

- 1) JavaScript – мова сценаріїв, або скриптів. Скрипт являє собою програмний код – набір інструкцій, який не вимагає попередньої обробки (наприклад, компіляції) перед запуском. Код JavaScript інтерпретується рушієм браузера під час завантаження веб-сторінки. Інтерпретатор браузера виконує порядковий

аналіз, обробку і виконання вихідної програми або запиту;

- 2) JavaScript – об'єктно-орієнтована мова з прототипним наслідуванням. Вона підтримує кілька вбудованих об'єктів, а також дозволяє створювати або видаляти свої власні (призначені для користувача) об'єкти. Об'єкти можуть наслідувати властивості безпосередньо один від одного, утворюючи ланцюжок об'єкт-прототип [20].

Структурно JavaScript можна представити у вигляді об'єднання трьох чітко помітних одна від одної частин:

- ядро (ECMAScript);
- об'єктна модель браузера (Browser Object Model або BOM);
- об'єктна модель документа (Document Object Model або DOM).

Якщо розглядати JavaScript в відмінних від браузера середовищах, то об'єктна модель браузера і об'єктна модель документа можуть не підтримуватися.

Об'єктну модель документа іноді розглядають як окрему від JavaScript сутність, що узгоджується з визначенням DOM як незалежного від мови інтерфейсу документа [21].

У браузері для JavaScript є все, що пов'язано з маніпулюванням веб-сторінками, взаємодією з користувачем і веб-сервером.

Наприклад, в браузері JavaScript може:

- додавати новий HTML-код на сторінку, змінювати існуючий вміст, модифікувати стилі;
- реагувати на дії користувача, клацання миші, перемістити вказівник, натискання клавіш;
- відправляти мережеві запити на віддалені сервера, завантажувати і завантажувати файли (технології AJAX і COMET);
- отримувати і встановлювати куки, задавати питання відвідувачеві, показувати повідомлення;
- запам'ятовувати дані на стороні клієнта ( «local storage»).

jQuery – бібліотека JavaScript, що містить в собі готові функції мови JavaScript, всі операції jQuery виконуються з коду JavaScript.

Бібліотека jQuery проводить маніпуляції з html-елементами, керуючи їх

поведінкою і використовуючи DOM для зміни структури веб-сторінки. При цьому вихідні файли HTML і CSS не змінюються, зміни вносяться лише в відображення сторінки для користувача.

Для вибору елементів використовуються селектори CSS. Вибір здійснюється за допомогою функції. При виклику функція повертає новий екземпляр об'єкта JQuery, який обертає нуль або більше елементів DOM і дозволяє взаємодіяти з ними різними способами.

Виконання різних сценаріїв можливо тільки після закінчення завантаження структури документа document, коли браузер перетворює HTML-код сторінки в дерево DOM. Управління процесом завантаження забезпечує конструкція [22].

### **3.4. Обґрунтування вибору бази даних**

В основному, існує два типи СУБД: реляційні та нереляційні, які також називаються SQL і NoSQL. Вони відрізняються за рівнем пошуку, розповсюдження та обробки даних.

Реляційні. Оскільки структурована мова запитів є ядром цих систем, цей тип також називається SQL. У реляційних СУБД дані відображаються у вигляді таблиць рядків та стовпців із суворою структурою та чіткими залежностями.

Завдяки інтегрованій структурі та системі зберігання даних, базам даних SQL не потрібні великі інженерні зусилля, щоб зробити їх добре захищеними. Вони є хорошим вибором для створення та підтримки складних програмних рішень, де будь-яка взаємодія має ряд наслідків. Однією з основ SQL є відповідність ACID (Atomicity, Consistency, Isolation, Durability). Відповідність ACID є кращим варіантом, якщо ви створюєте, наприклад, електронну комерцію або фінансові програми, де цілісність бази даних є критичною.

Однак масштабування може бути складним завданням для баз даних SQL. Масштабування бази даних SQL між декількома серверами (горизонтальне масштабування) вимагає додаткових зусиль інженерії. Натомість бази даних SQL зазвичай масштабуються вертикально, тобто додаючи більше обчислювальної

потужності на сервер.

Приклади баз даних SQL:

- MySQL;
- MariaDB;
- Oracle;
- PostgreSQL;
- MSSQL.

Нереляційні. Оскільки ці бази даних не обмежені структурою таблиці, вони називаються NoSQL. Цей тип системи управління базами даних вважається орієнтованим на документи. Неструктуровані дані, такі як статті, фотографії, відеоролики та ін., збираються в одному документі. Дані прості для запиту, але не завжди класифікуються у рядки та стовпці, як у реляційній базі даних. Нереляційні або бази даних NoSQL зазвичай горизонтально масштабуються шляхом додавання серверів.

Оскільки бази даних NoSQL дозволяють резервувати різні типи даних разом і масштабувати їх, зростаючи навколо декількох серверів, їх популярність не зменшується, що є цілком зрозумілим. Крім того, створення MVP - це чудовий варіант для стартапів із спритневою розробкою Agile. NoSQL не вимагає підготовки до розгортання, що полегшує швидке оновлення структури даних без часових відстань.

Існують такі NoSQL бази даних:

- MongoDB;
- Redis;
- Cassandra;
- Elasticsearch.

Оскільки для розробки системи не потрібно зберігати у базі складні структури даних, а масштабування бази не є пріоритетом, адже набір даних буде досить обмеженим, було вирішено обрати SQL базу даних.

Далі постає питання обрання SQL бази з існуючих. Для порівняння було обрано бази PostgreSQL, MySQL та MSSQL.

Переваги PostgreSQL:

- Масштабованість. Вертикальна масштабованість є відмітною ознакою PostgreSQL, на відміну від MySQL СУБД. Враховуючи, що практично будь-яке користувацьке програмне рішення має тенденцію до зростання, що призводить до розширення бази даних, цей конкретний варіант, безумовно, підтримує зростання та розвиток бізнесу [23].
- Підтримка користувацьких типів даних. PostgreSQL за замовчуванням підтримує велику кількість типів даних, таких як JSON, XML, H-Store та інші. PostgreSQL цим користується, будучи однією з небагатьох реляційних баз даних з сильною підтримкою функцій NoSQL. Крім того, це дозволяє користувачам визначати власні типи даних. Оскільки система може потребувати різних типів баз даних протягом усього свого існування для кращої продуктивності, ця опція приносить більшу гнучкість таблиці.
- Легко інтегрує сторонні інструменти. Система управління базами даних PostgreSQL підтримує додаткові інструменти, як безкоштовні, так і комерційні. Сфера застосування включає розширення для покращення багатьох аспектів. Наприклад, ClusterControl надає вражаючу допомогу в управлінні, моніторингу та масштабуванні баз даних SQL і NoSQL з відкритим кодом.
- Відкритий код і велика спільнота. Postgres має повністю відкритий код та підтримується його спільнотою, що зміцнює його як повноцінну екосистему. Крім того, розробники завжди можуть очікувати безкоштовної та швидкої допомоги громади.

#### Недоліки PostgreSQL:

- Невідповідна документація. У той час як PostgreSQL має велике співтовариство і надає потужну підтримку його учасникам, у документації все ще бракує послідовності та повноти. Оскільки PostgreSQL-спільнота досить розповсюджена, документація не відповідає рівним стандартам для всіх функцій Postgre.
- Відсутність інструментів звітності та аудиту. Істотним недоліком PostgreSQL є відсутність інструментів перегляду, які б показували поточний

стан бази даних. Вам доведеться постійно перевіряти, чи щось не так. Завжди існує ризик, що інженери БД помітять помилку занадто пізно.

Розглянемо основні переваги MySQL:

- Безкоштовне завантаження. Видання MySQL для спільноти можна безкоштовно завантажити. Звичайно, є й інші, передплачені варіанти для цілей Enterprise або Cluster із більш багатим функціоналом. Тим не менш, якщо для некомерційних або освітніх проєктів, модель безкоштовного завантаження є найбільш підходящою для нового запуску [24].
- Простий синтаксис і невелика складність. Структура та стиль MySQL дуже прості. Розробники навіть вважають MySQL базою даних із людською мовою. Оскільки MySQL часто використовується в тандемі з мовою програмування PHP. Оскільки вони мають легку криву навчання, вам не потрібно буде наймати кваліфікованого розробника для управління вашою базою даних. Також MySQL простий у використанні.
- Сумісна із хмарними обчисленнями. Бізнес-орієнтована за своєю природою і спочатку розроблена для Інтернету, MySQL підтримується найпопулярнішими хмарними провайдерами. Вона доступна на таких провідних платформах, як Amazon, Microsoft та інші. Це робить MySQL ще більш привабливою і надає підприємствам, що використовують його, місце для зростання [25].

Недоліки MySQL:

- Проблеми масштабування. MySQL не був побудований враховуючи масштабування. MySQL можна масштабувати MySQL, але для цього знадобиться більше зусиль для інженерів порівняно з будь-якою з баз даних NoSQL [26].
- Частково відкритий код. Хоча MySQL має частину з відкритим кодом, вона здебільшого має ліцензію Oracle. Це обмежує спільноту MySQL з точки зору вдосконалення СУБД.
- Обмежена відповідність стандартам SQL. Структурована мова запитів має конкретні стандарти [25]. MySQL не повністю виконує їх, тобто MySQL не

підтримує деякі стандартні функції SQL. З іншого боку, MySQL має деякі розширення та чіткі функції, які не відповідають стандартам структурованої мови запитів.

Останньою розглянемо MSSQL.

Переваги MSSQL:

- Різноманітність версій. Microsoft SQL Server надає широкий вибір різних варіантів з різноманітними функціями [27]. Видання Express із безкоштовною базою даних пропонує інструменти початкового рівня, ідеальне поєднання для навчання та створення настільних або невеликих серверних додатків, керованих даними. Опція Developers дозволяє створювати та тестувати додатки, включаючи деякі функціональні можливості підприємства, але без ліцензії на виробничий сервер. Для великих проектів існують також веб-, стандартні та корпоративні видання з різною мірою адміністративних можливостей та рівнів обслуговування.
- Цільове рішення для бізнес-даних. Орієнтуючись на переважно комерційні рішення, MSSQL надає безліч функцій з додатковою вартістю бізнесу. Необов'язковий вибір компонентів дозволяє будувати рішення ETL, формувати базу знань та здійснювати очищення даних [27]. Крім того, він пропонує інструменти для загального адміністрування даних, он-лайн аналітичної обробки та обміну даними, додатково надаючи варіанти для створення звітів та візуалізації.
- Докладна документація та допомога громади. Завдяки тому, що Microsoft SQL Server спрямоване на комплексне обслуговування баз даних, повна онлайн-документація також відображає цю концепцію. Структуровані вказівки, численні відомості та демонстрації демонструють повну картину в системі даних MSSQL. Microsoft Premier надає доступ до спеціалізованої підтримки спільноти Microsoft [20].

Недоліки MSSQL:

- Підтримка хмарних баз даних. Будучи частиною послідовної екосистеми Майкрософт, MSSQL може бути інтегрований з хмарою Microsoft, базами



даних SQL Azure або SQL сервером на віртуальних машинах Azure. Рішення дозволяють перенести адміністрування баз даних у хмару, якщо база даних вашого програмного забезпечення стає дійсно великою і важкою для адміністрування [27].

- Велика ціна. Здебільшого використовується в масштабах підприємства, MSSQL Server залишається одним з найдорожчих рішень.
- Часта зміна ліцензійних умов. Інша проблема - це постійно змінюється процес ліцензування. Саму стратегію ціноутворення важко зрозуміти, і елементи, що входять до конкретного видання, змінюються, переходячи з одного тарифного плану у інший
- Складний процес налаштування. Для початківців, яким доводиться керувати великими наборами даних, робота з оптимізацією запитів та налаштуванням продуктивності може бути проблематичною. Оскільки процес не такий очевидний, він може створити значні проблеми на початку розробки [28].

Проаналізувавши всі наведені вище переваги і недоліки баз даних було вирішено обрати для розробки системи СУБД PostgreSQL. Основним недоліком PostgreSQL є відсутність належної документації, але завдяки Django ORM ця проблема нівелюється.

## 4. ОПИС ПРОГРАМНОЇ РЕАЛІЗАЦІЇ

Розроблена система ранжирування територій України за рівнем соціально-екологічних ризиків людського розвитку складається з декількох частин, кожна з яких спроектована і реалізована із використанням сучасних засобів розробки.

### 4.1. Архітектура програмної системи

Програма має трирівневу архітектуру та складається серверної та клієнтської частини (рисунок 4.1).

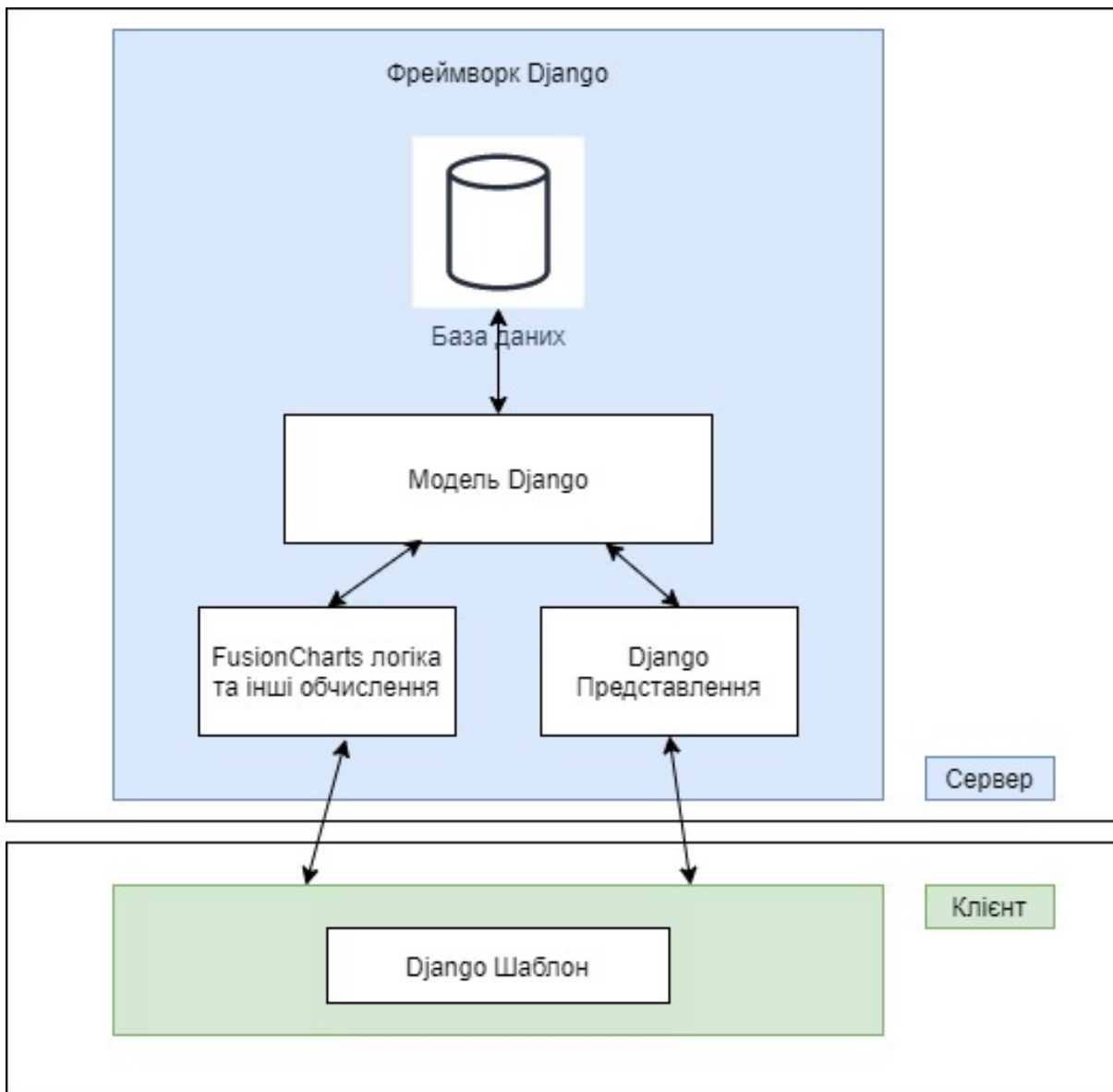


Рисунок 4.1 – Архітектура системи

Серверна частина складається з Баз Даних, Django Model та Django View. Model інкапсулює взаємодію з базою даних за допомогою ORM (Object-Relational Mapping) технології. Після створення моделі, Django автоматично створює API для роботи з базою даних, який дозволяє створювати, одержувати, змінювати і видаляти об'єкти.

Django View відповідає за обробку та перетворення отриманих із шару доступу даних. У ньому виконуються всі необхідні обчислення, після чого об'єкти передаються у на рівень представлення даних (Client).

Client відповідає за всі об'єкти, з якими взаємодіє користувач. На цьому рівні знаходяться всі компоненти, з яких складається користувацький інтерфейс (Django template, стилі, статичні сторінки HTML, JavaScript), а також моделі представлень, контролери, об'єкти контексту запиту.

- Система на базі Django-фреймворку розроблена з особливою підтримкою “слабких зв'язків” і строгим розмежуванням складових частин програми. Разом ці три частини – логіка доступу до даних, логіка виконання і логіка надання даних – складають концепт, який зазвичай називають концепцією Model-View-Controller (MVC) (модель-уявлення-контролер) архітектури додатків.

- Модель – частина доступу до даних, обслуговується системою баз даних Django;

- Представлення – частина, яка займається вибіркою того, що саме і як саме відображати, в Django цим займаються уявлення (views) і шаблони (templates);

- Контролер – частина, яка визначає представлення в залежності від вказівок користувача, цим займається сам фреймворк, слідуючи налаштуванням URLconf і викликаючи відповідну функцію Python для заданого URL-а.

Схема взаємодії зображена на рисунку 4.2.



Рисунок 4.2 – Принцип взаємодії MVC

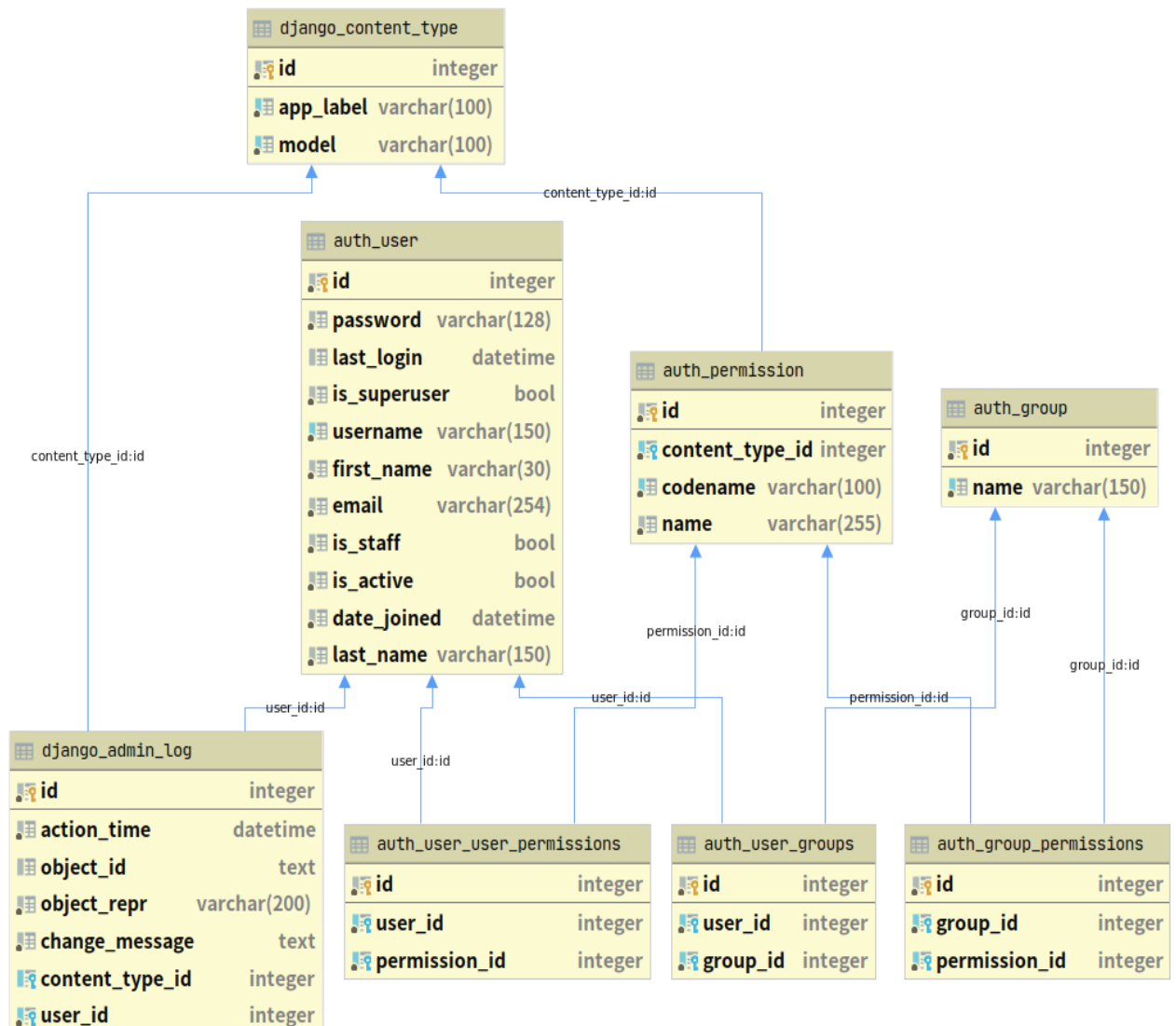
Побудова графіків і карт відбувається за допомогою бібліотеки FusionCharts.

## 4.2. Опис бази даних

В процесі роботи з системою для побудови графіків використовуються дані, завантажені з бази даних. На рисунку 4.3 зображена схема таблиць бази даних, що відповідає за права користувача і інформацію про нього. Дані таблиці згенеровані за допомогою веб-фреймворка Django.

До користувацьких належать наступні таблиці:

- “django\_content\_type” – містить запис для кожної встановленої моделі;
- “auth\_user” – містить список усіх користувачів в системі;
- “auth\_permission” – таблиця можливих дозволів;
- “auth\_group” – таблиця можливих груп користувачів;
- “django\_admin\_log” – журнали помилок, згенерованих у веб-системі, моделлю Адміністратор;
- “auth\_user\_user\_permissions” – таблиця із записами про права кожного користувача;
- “auth\_user\_groups” – таблиця, що містить записи із всіма групами користувачів;
- “auth\_group\_permissions” – таблиця із записами дозволів для кожної групи користувачів.



Таблиця “auth\_user” (таблиця 4.1) містить список усіх користувачів в системі. Після реєстрації, користувач може змінювати значення полів email, first\_name, last\_name. Інші поля таблиці “auth\_user” може редагувати тільки адміністратор.

Назва поля	Тип поля	Опис
id	int	Первинний ключ
username	varchar	Унікальне ім'я користувача
email	email	Email адреса
first_name	varchar	Ім'я користувача
last_name	varchar	Прізвище користувача

Таблиця 4.1. Структура таблиці “auth\_user” (Продовження)

is_staff	bool	Чи є користувач персоналом
is_active	bool	Чи є користувач активним
is_superuser	bool	Чи є користувач супер'юзером
date_joined	datetime	Дата та час реєстрації користувача
last_login	datetime	Дата та час входу в систему користувачем

На рисунку 4.4 показано приклад заповнення таблиці “auth\_user”.

Виберіть користувач щоб змінити

Дія: -----  0 з 2 обрано

<input type="checkbox"/>	ІМ'Я КОРИСТУВАЧА	EMAIL АДРЕСА	ІМ'Я	ПРІЗВИЩЕ	СТАТУС ПЕРСОНАЛУ
<input type="checkbox"/>	admin	admin@example.com			✓
<input type="checkbox"/>	amita051197@gmail.com				✗

2 користувачі

Рисунок 4.4 – Приклад заповнення таблиці “auth\_user”

На рисунку 4.5 зображена схема таблиць, що відповідають за збереження даних для побудови графіків. Ці таблиці відповідають за збереження даних, що адміністратор завантажує з сайту державної статистики України і додає вручну.

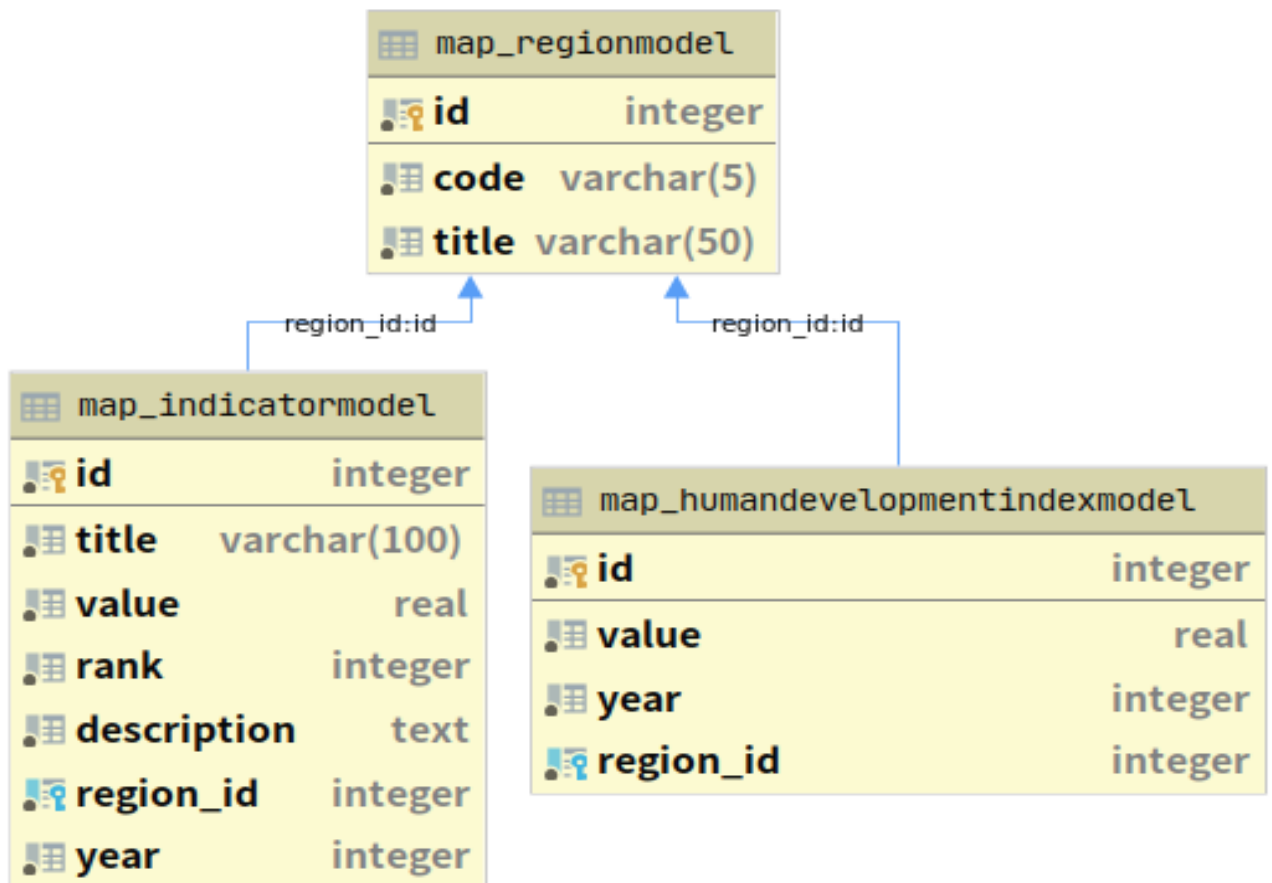


Рисунок 4.5 – Схема таблиц показників

Таблиця “map\_regionmodel” (таблиця 4.2) відповідає за список всіх областей України.

Таблиця 4.2. Структура таблиці “map\_regionmodel”

Назва поля	Тип поля	Опис
id	int	Первинний ключ
code	varchar	Код області
title	varchar	Назва області

На рисунку 4.6 наведено приклад заповнення таблиці “map\_regionmodel”.

Виберіть region model щоб змінити

Дія:   0 з 27 обрано

- ☐ REGION MODEL
- ☐ АР Крим
- ☐ Вінницька
- ☐ Волинська
- ☐ Дніпропетровська
- ☐ Донецька
- ☐ Житомирська
- ☐ Закарпатська
- ☐ Запорізька
- ☐ Івано-Франківська
- ☐ Київ
- ☐ Київська
- ☐ Кіровоградська
- ☐ Луганська
- ☐ Львівська
- ☐ Миколаївська
- ☐ Одеська
- ☐ Полтавська
- ☐ Рівненська
- ☐ Севастопіль

Рисунок 4.6 – Приклад заповнення таблиці “map\_regionmodel”

Таблиця “map\_humandevlopmentindexmodel” (таблиця 4.3) містить список інтегральних індексів людського розвитку по кожному регіону за кожен рік.

Таблиця 4.3. Структура таблиці “map\_humandevlopmentindexmodel”

Назва поля	Тип поля	Опис
id	int	Первинний ключ
value	real	Значення показника
year	int	Рік
region_id	int	Id області

Таблиця “map\_humandevlopmentindexmodel” може заповнюватися тільки адміністратором сайту. На рисунку 4.7 наведено приклад заповнення даної таблиці.



Виберіть human development index model щоб змінити

Дік:  Вперед 0 з 100 обрано

- ☐ HUMAN DEVELOPMENT INDEX MODEL
- ☐ 2004, Чернігівська, 3.1473
- ☐ 2004, Чернівецька, 3.5332
- ☐ 2004, Черкаська, 3.4433
- ☐ 2004, Хмельницька, 3.3595
- ☐ 2004, Херсонська, 3.284
- ☐ 2004, Харківська, 3.4865
- ☐ 2004, Тернопільська, 3.4707
- ☐ 2004, Сумська, 3.2559
- ☐ 2004, Рівненська, 3.3522
- ☐ 2004, Полтавська, 3.5373
- ☐ 2004, Одеська, 3.3546
- ☐ 2004, Миколаївська, 3.423
- ☐ 2004, Львівська, 3.5408
- ☐ 2004, Луганська, 3.1709
- ☐ 2004, Кіровоградська, 3.2075
- ☐ 2004, Київська, 3.5052
- ☐ 2004, Івано-Франківська, 3.3833
- ☐ 2004, Запорізька, 3.3693
- ☐ 2004, Закарпатська, 3.7834
- ☐ 2004, Житомирська, 3.261
- ☐ 2004, Донецька, 3.4375
- ☐ 2004, Дніпропетровська, 3.4776
- ☐ 2004, Волинська, 3.3317
- ☐ 2004, Вінницька, 3.3848

Рисунок 4.7 – Приклад заповнення таблиці “map\_humandevelopmentindexmodel”

Таблиця "map\_indicatoremodel" (таблиця 4.4) містить список значень кожного з аспектів людського розвитку по кожному регіону за кожен рік.

Таблиця 4.4. Структура таблиці “map\_indicatoremodel”

Назва поля	Тип поля	Опис
id	int	Первинний ключ

Таблиця 4.4. Структура таблиці “map\_indicatoremodel” (Продовження)

value	real	Значення показника
year	int	Рік
rank	int	Ранг показника
region_id	int	Id області
description	text	Опис показника розвитку
title	varchar	Назва показника

Таблиця “map\_indicatoremodel” також може заповнюватися тільки адміністратором сайту. На рисунку 4.8 наведено приклад заповнення даної таблиці.

☐ INDICATOR MODEL

☐ Відтворення населення, Чернігівська, 2017, 0.6895

☐ Відтворення населення, Чернівецька, 2017, 0.6866

☐ Відтворення населення, Черкаська, 2017, 0.6697

☐ Відтворення населення, Хмельницька, 2017, 0.672

☐ Відтворення населення, Херсонська, 2017, 0.6692

☐ Відтворення населення, Харківська, 2017, 0.6683

☐ Відтворення населення, Тернопільська, 2017, 0.6817

☐ Відтворення населення, Рівненська, 2017, 0.6981

☐ Відтворення населення, Полтавська, 2017, 0.7725

☐ Відтворення населення, Одеська, 2017, 0.6739

☐ Відтворення населення, Миколаївська, 2017, 0.7072

☐ Відтворення населення, Львівська, 2017, 0.6906

☐ Відтворення населення, Кіровоградська, 2017, 0.6189

☐ Відтворення населення, Київська, 2017, 0.7281

☐ Відтворення населення, Івано-Франківська, 2017, 0.7099

☐ Відтворення населення, Запорізька, 2017, 0.6831

☐ Відтворення населення, Закарпатська, 2017, 0.6571

☐ Відтворення населення, Житомирська, 2017, 0.6769

☐ Відтворення населення, Дніпропетровська, 2017, 0.664

☐ Відтворення населення, Волинська, 2017, 0.718

☐ Відтворення населення, Вінницька, 2017, 0.6966

Рисунок 4.8 – Приклад заповнення таблиці “map\_indicatoremodel”

Отже, для збереження статистичних даних інтегрального індексу людського розвитку та інформації по кожний з чинників людського розвитку окремо було створено таблиці “map\_indicatoremodel”, “map\_humandevlopmentindexmodel” та “map\_regionmodel”.

Для доступу до бази даних використовується Django ORM. Створюється клас сутності мовою Python описом всіх полів таблиці. Даний клас дозволить звертатися до таблиці як до об’єкту Python. Класи мають назви IndicatorModel, HumanDevelopmentIndexModel та RegionModel відповідно.

## **5. МЕТОДИКА РОБОТИ КОРИСТУВАЧА З ПРОГРАМНОЮ СИСТЕМОЮ**

Для забезпечення безвідмовної роботи програмної системи ранжирування регіонів України за рівнем соціально-екологічних ризиків людського розвитку необхідно дотримуватися основних рекомендацій щодо її використання.

### **5.1. Системні вимоги та інсталяція програмного продукту**

Веб застосунок працює на віддаленому сервері, доступному в інтернеті, тому не потребує інсталяції. Розроблена система працює на веб-сервері, що є доступним в інтернеті. Також система може бути запущена локально, для цього на комп'ютер має бути встановлена мова програмування Python версії 3.7.6, всі пакети, що описані в Pipfile та база даних PostgreSQL.

Рекомендовані вимоги до системи користувача:

- доступ до мережі Інтернет зі швидкістю не менше 1Мбіт/сек;
- об'єм оперативної пам'яті 1 Гб та більше;
- браузер Chrome, Opera, Mozilla Firefox тощо з підтримкою cookie та JavaScript (браузер бажано оновити до останньої версії).

### **5.2. Сценарії роботи користувача з системою**

Для того, щоб відкрити застосунок локально, потрібно запустити Django сервер на комп'ютері або перейти на сайт за посиланням. Після цього відкривається головна сторінка як зображено на рисунку 5.1, на якій написана назва проекту та короткий опис його можливостей.

# Система ранжування територій України за рівнем соціально-екологічних ризиків

Вітаємо на сайті, admin!

Рисунок 5.1 — Головна сторінка

На кожній сторінці сайту зверху відображається панель навігації (рисунок 5.2), на якій розміщені назви всіх доступних модулів програми.

Рисунок 5.2 — Навігаційна панель

Якщо користувач не авторизований в системі, для нього доступні опції реєстрації і авторизації. Для того, щоб зареєструватися на сайті, користувач повинен ввести унікальне ім'я, персональний пароль, підтвердити його та вказати електронну пошту (рисунок 5.3).

## Реєстрація

Ім'я користувача:

Пароль:

Підтвердження пароля:

Email:

Зареєструватися

Рисунок 5.3 – Форма реєстрації

Якщо користувач ввів вже існуюче ім'я або занадто легкий пароль, після натиснення кнопки "Зареєструватися" біля кожного з полів, яке було неправильно заповнене, виникне повідомлення про помилку і реєстрація не відбудеться (рисунок 5.4). Для того, щоб створити нового користувача, вся форма має бути заповнена правильно.

The image shows a web registration form titled "Реєстрація" (Registration). It contains several input fields and error messages. The "Ім'я користувача:" (Username) field contains the text "admin". The "Пароль:" (Password) field is empty. Below the password field, there are two red error messages: "Пароль надто короткий. Він повинен містити як мінімум 8 символів" (Password is too short. It must contain at least 8 symbols) and "Пароль надто відомий." (Password is too common). Below these, the "Підтвердження пароля:" (Confirm password) field is empty. The "Email:" field contains the text "test@gmail.com". At the bottom of the form is a button labeled "Зареєструватися" (Register).

Реєстрація

Користувач з таким ім'ям вже існує.

Ім'я користувача:

admin

Пароль:

Пароль надто короткий. Він повинен містити як мінімум 8 символів

Пароль надто відомий.

Підтвердження пароля:

Email:

test@gmail.com

Зареєструватися

Рисунок 5.4 – Помилки при реєстрації

Після реєстрації користувач повинен авторизуватися в системі, для цього необхідно вибрати на навігаційній панелі опцію "Вхід". З'явиться форма авторизації (рисунок 5.5). Як у випадку з реєстрацією, користувач має правильно ввести унікальне ім'я, під яким він зареєструвався і свій пароль.

**Вхід**

Ім'я користувача:

Пароль:

**Увійти**

Рисунок 5.5 – Форма входу в систему

Також кожному користувачеві доступна форма зв'язку з розробником (рисунок 5.6). У ній користувач має вказати свою електронну пошту, тему повідомлення та його зміст.

**Зв'яжіться з нами**

Ваш e-mail:

Тема:

Повідомлення:

**Надіслати**

Рисунок 5.6 – Форма зв'язку з розробником

Щоб перейти на сторінку з картою регіонів України, необхідно на панелі вибрати “Інтегральний індекс людського розвитку” та з додаткового меню, обрати рік, за який бажано отримати дані (рисунок 5.7).

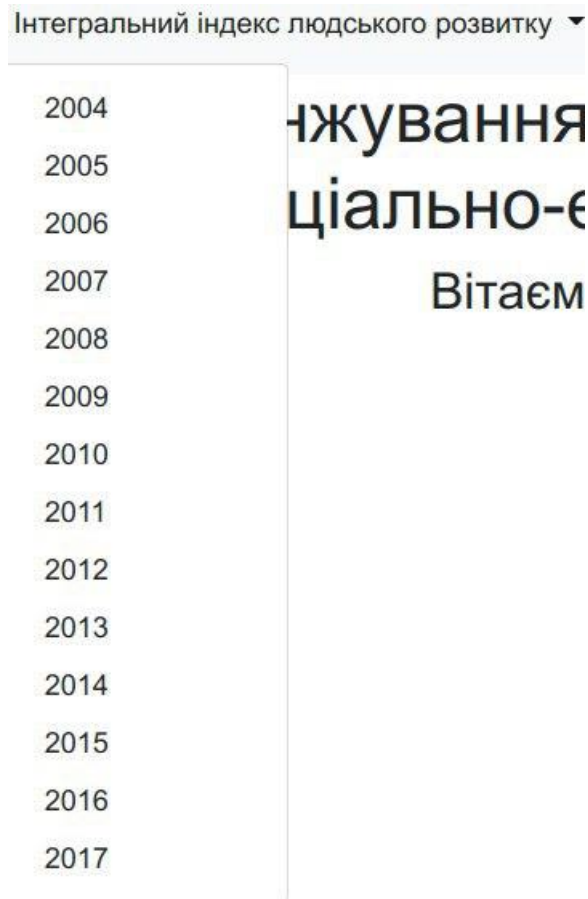


Рисунок 5.7 – Меню вибору року

На сторінці “Індекс людського розвитку” розміщена інтерактивна карта України (рисунок 5.8), на кожному регіоні якої відображене значення інтегрального показника по регіону та код області, а при наведенні на область з’являється повна її назва. Кольори на карті відповідають за рівень індексу людського розвитку у відповідності до загальноприйнятих показників. Кольори змінюються від темно-червоного, для найнижчого значення, до зеленого, для найвищого. Таке ранжування дозволяє наглядно продемонструвати рівень людського розвитку у кожному регіоні і на основі цього зробити певні висновки відносно ризиків людського розвитку.



Нижче карти розташована панель управління картою, що розбита на кольори за значенням показника інтегрального розвитку регіону (рисунок 5.9) від найменшого значення, якому відповідає червоний колір, до найвищого, якому відповідає зелений.

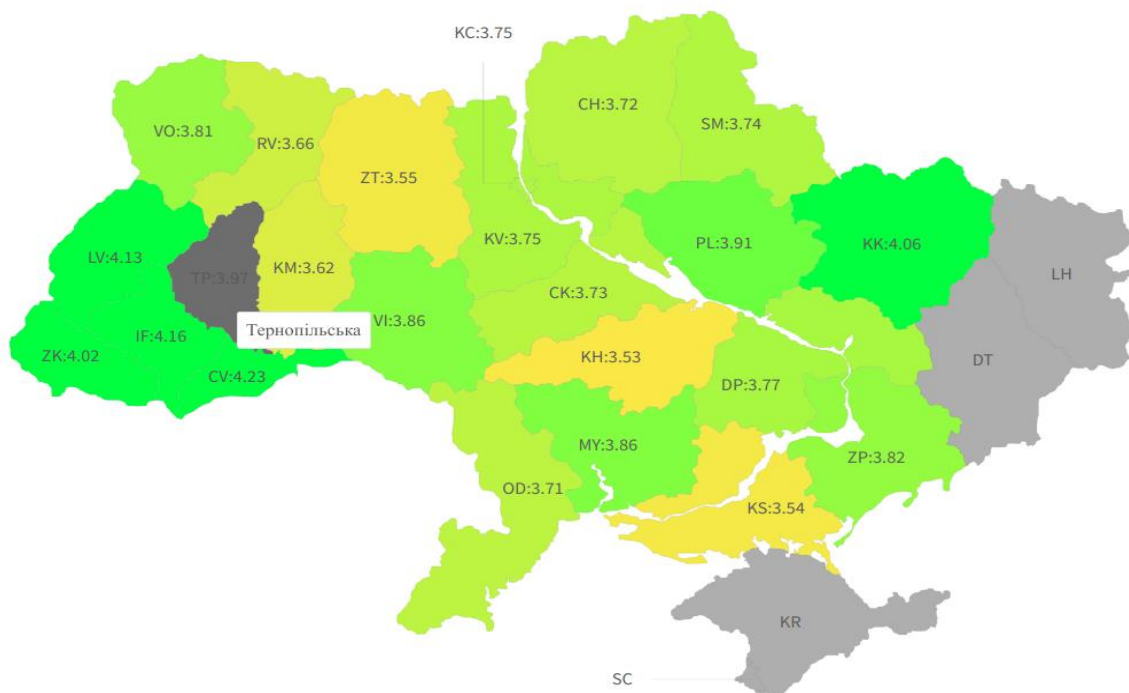


Рисунок 5.8 – Інтерактивна карта “Індекс людського розвитку”



Рисунок 5.9 – Панель управління картою

Зліва і з права знаходяться повзунки, перетягуючи які можна залишати кольоровими тільки ті регіони, які входять в обраний діапазон значень, решта ж стає білими (рисунок 5.10).

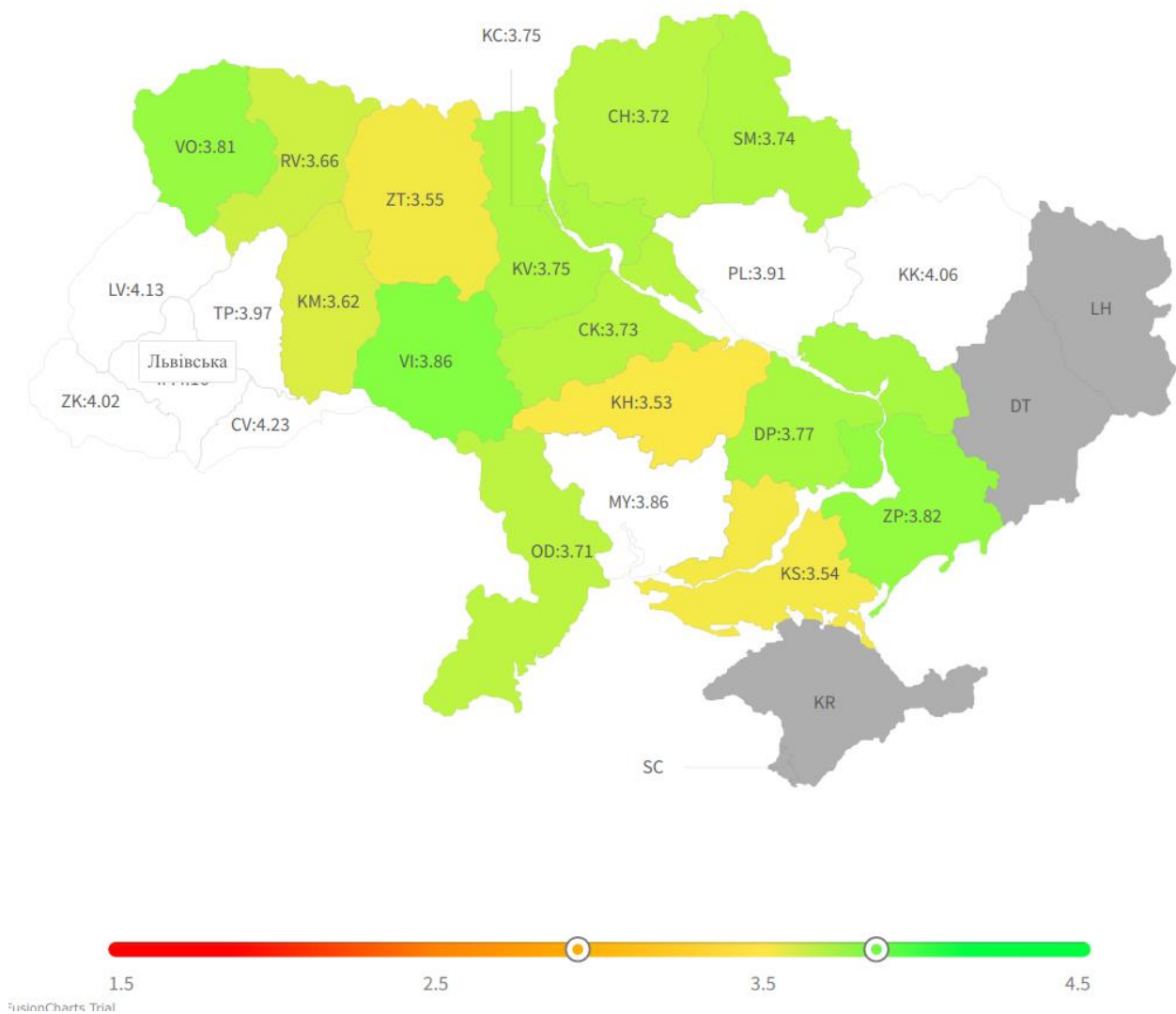


Рисунок 5.10 – Вибір певного діапазону значень

Сірим виділені ті регіони, дані по яких за обраний рік відсутні. Якщо натиснути мишкою на обраний регіон, то користувач перейде за посиланням на сторінку регіону, де відображений графік зміни значення інтегрального індексу людського розвитку по обраному регіону (рисунок 5.11).

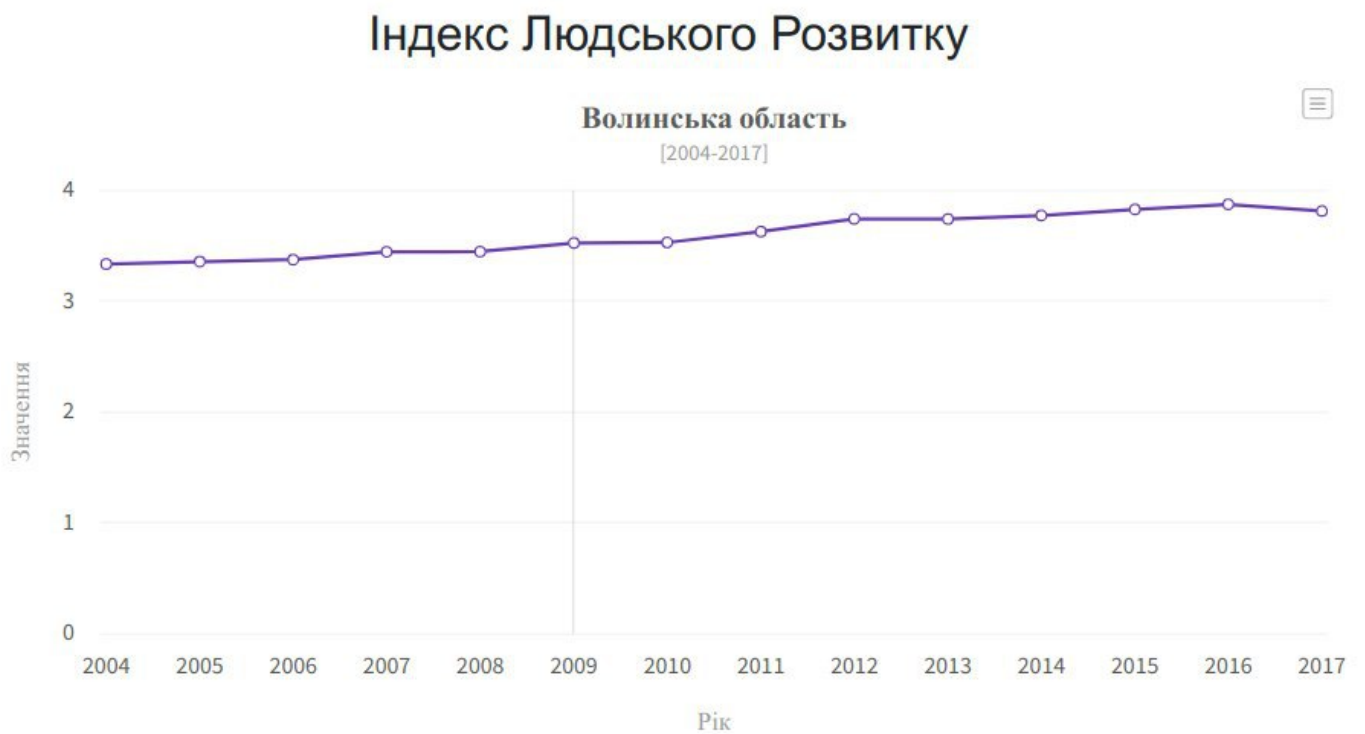


Рисунок 5.11 – Графік зміни індексу людського розвитку по регіону

Для того, щоб перейти на сторінку чинників людського розвитку, необхідно вибрати “Чинники людського розвитку”. Із списку потрібно вибрати бажаний чинник (рисунок 5.12).

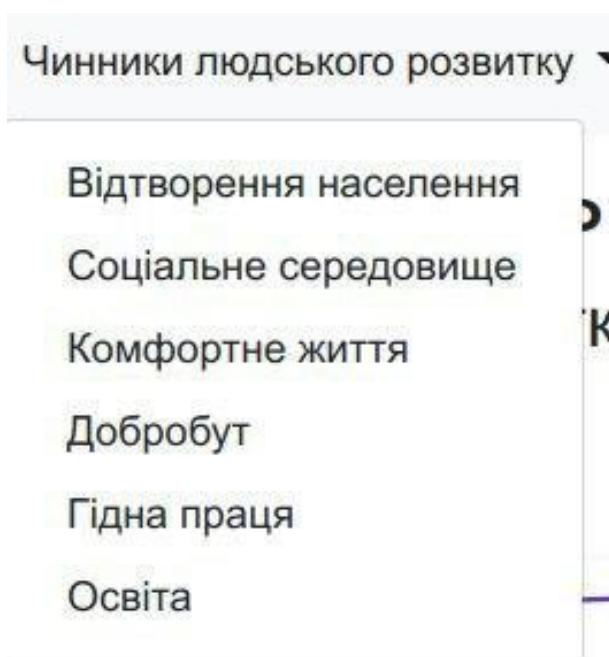


Рисунок 5.12 – Список чинників людського розвитку

При переході на сторінку обраного чинника, користувач може побачити інтерактивну карту України за певний рік, на якій зображено числові значення чинника по регіонах (аналогічно до сторінки “Інтегральний індекс людського розвитку”) (рисунок 5.13).

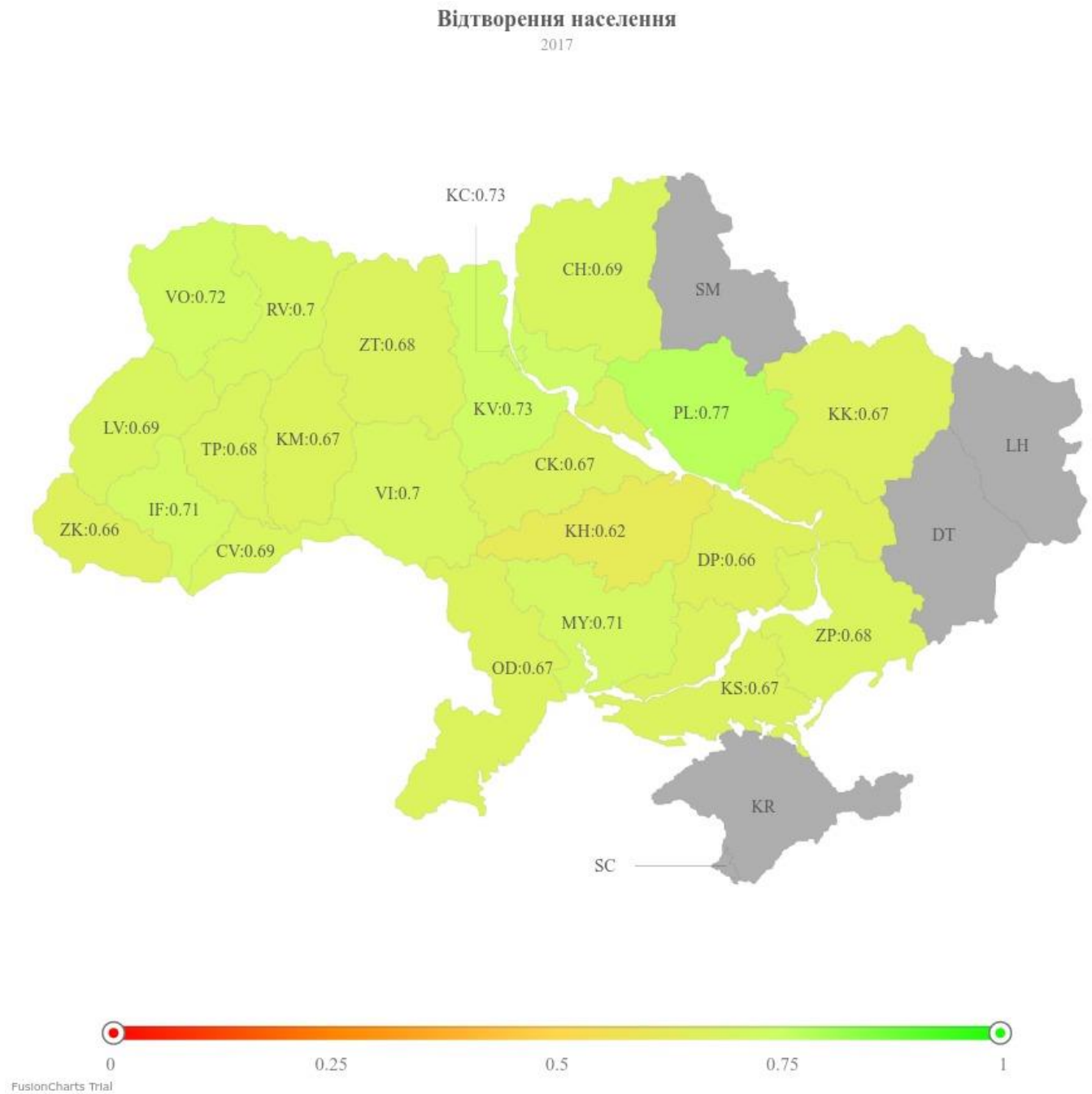


Рисунок 5.13 – Інтерактивна карта “Відтворення населення” за 2017 рік

При переході на сторінку регіону можна переглядати діаграми зміни чинника впродовж доступного періоду. Діаграми двох типів: лінійна (рисунок 5.14) та стовпчикова (рисунок 5.15).

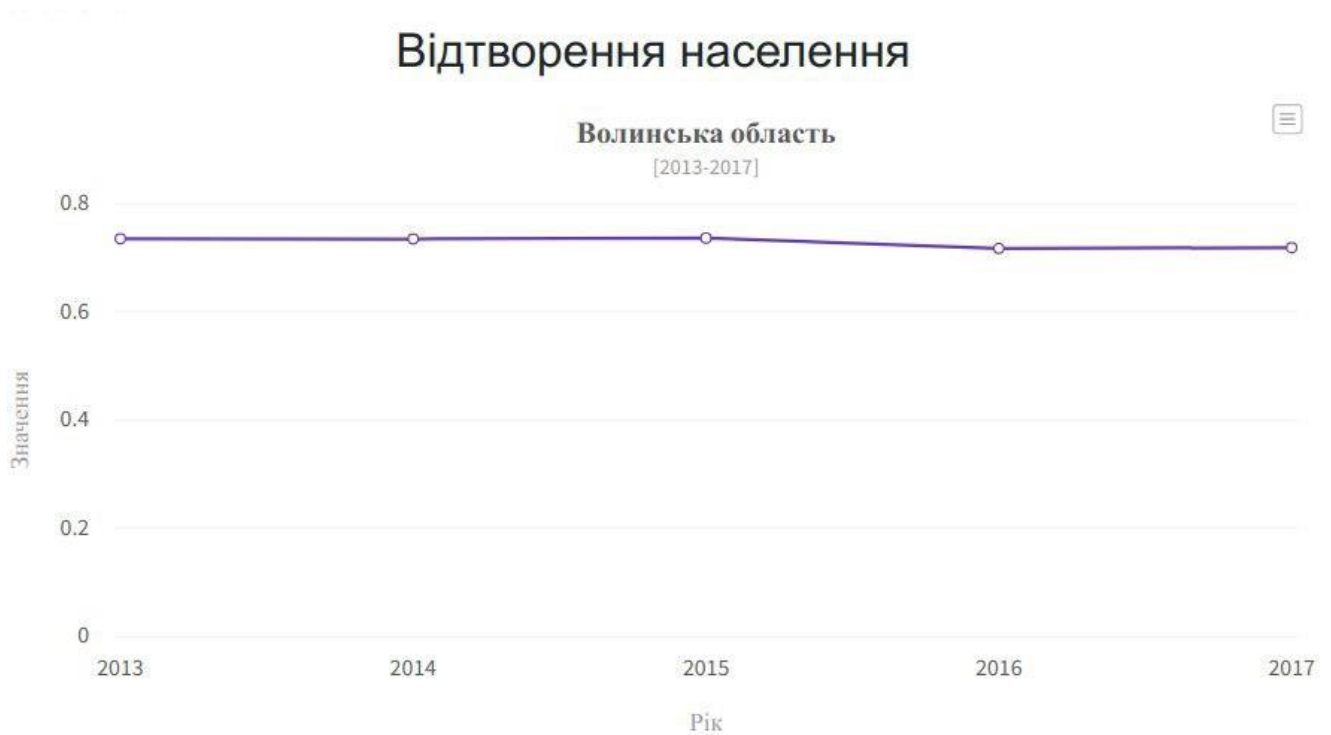


Рисунок 5.14 – Лінійна діаграма “Відтворення населення”

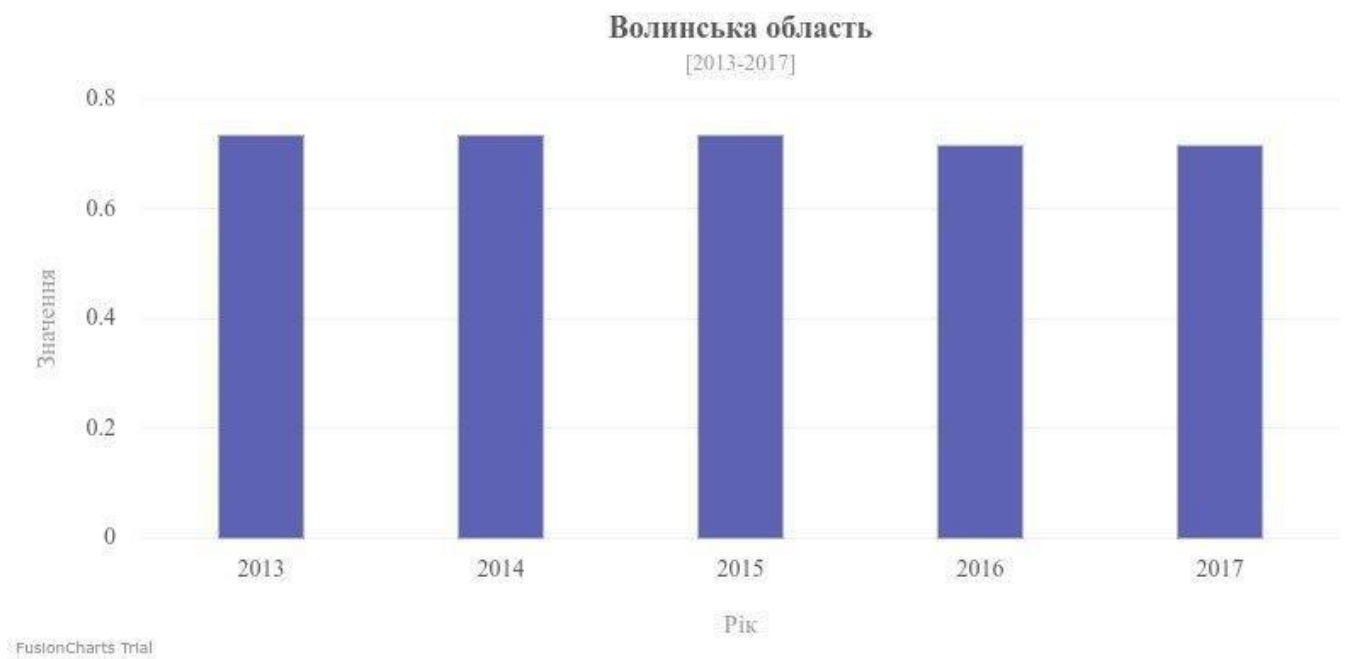


Рисунок 5.15 – Стовпчикова діаграма “Відтворення населення”

Кожен з побудованих графіків має позначку, що зображена на рисунку 5.16. При наведенні на неї з'являється список усіх можливих форматів для збереження, а саме [29]:

- PNG, Portable Network Graphics – це зображення, що містить растрові карти, стиснуті без за технологією стиснення без втрат. Файли PNG зазвичай використовуються для зберігання веб-графіки, цифрових фотографій та зображень із прозорим фоном;
- JPG, Joint Photographic Experts Group – зображення, збережене у форматі стисненого зображення, стандартизованому Joint Photographic Experts Group (JPEG). Він зазвичай використовується для зберігання цифрових фотографій і використовується більшістю цифрових камер для збереження зображень;
- PDF, Portable Document Format File – це багатоплатформенний документ, створений Adobe Acrobat або іншою програмою PDF. Формат PDF зазвичай використовується для збереження документів і публікацій у стандартному форматі, який можна переглядати на багатьох платформах;
- SVG, Scalable Vector Graphics File – це графічний файл, який використовує двовимірний векторний графічний формат, створений консорціумом всесвітньої веб-мережі (W3C, World Wide Web Consortium). Він описує зображення у текстовому форматі, який базується на XML. Файли SVG розробляються як стандартний формат для відображення векторної графіки в Інтернеті;
- CSV, Comma Separated Values – це файл значень, відокремлений комами, зазвичай використовується програмами електронних таблиць, такими як Microsoft Excel або OpenOffice Calc. Він містить набори простих текстових даних, розділених комами з кожним новим рядком у файлі CSV, що представляє нову рядок бази даних та кожен рядок бази даних, що складається з одного або декількох полів, розділених комою;
- XLSX, Microsoft Excel Open XML Spreadsheet – це таблиця Excel, створена Microsoft Excel або іншою програмою електронних таблиць, наприклад OpenOffice Calc або Apple Numbers. Він зберігає дані у робочих аркушах, які містять комірки, розташовані в сітці рядків та стовпців, а також можуть містити діаграми, математичні функції, стилі та форматування.



Рисунок 5.16 – Збереження графіка на пристрій

На рисунку 5.17 наведено приклад збереження графіка Індекс Регіонального Розвитку України у форматі PNG.

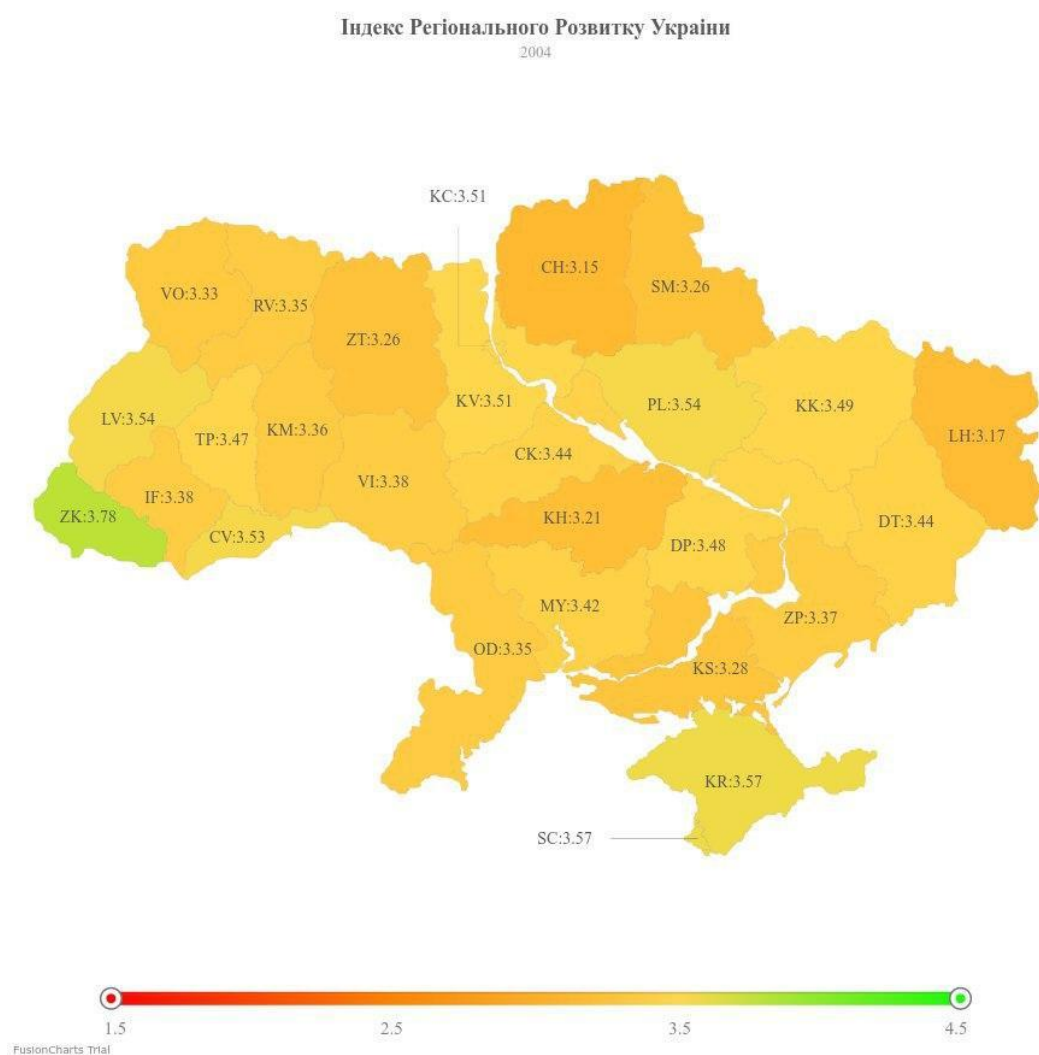


Рисунок 5.17 – Графік, збережений у форматі PNG



Оскільки користувачеві можуть знадобитися не лише рисунки, а і табличні дані, графіки можна зберігати у форматах CSV або XLSX (рисунок 5.18).




<b>Id</b>	<b>Short Name</b>	<b>Long Name</b>	<b>Value</b>	<b>Formatted Value</b>
ua.sc	SC	Sevastopol	3,5693	3,57
ua.kc	KC	Kiev	3,5052	3,51
ua.kr	KR	Crimea	3,5693	3,57
ua.kv	KV	Kiev Oblast	3,5052	3,51
ua.ch	CH	Chernihiv	3,1473	3,15
ua.pl	PL	Poltava	3,5373	3,54
ua.ck	CK	Cherkasy	3,4433	3,44
ua.kh	KH	Kirovohrad	3,2075	3,21
ua.dp	DP	Dnipropetrovsk	3,4776	3,48
ua.zp	ZP	Zaporizhia	3,3693	3,37
ua.ks	KS	Kherson	3,284	3,28
ua.my	MY	Mykolaiv	3,423	3,42
ua.lh	LH	Luhansk	3,1709	3,17
ua.dt	DT	Donetsk	3,4375	3,44
ua.od	OD	Odessa	3,3546	3,35
ua.vi	VI	Vinnytsia	3,3848	3,38
ua.cv	CV	Chernivtsi	3,5332	3,53
ua.zk	ZK	Zakarpattia	3,7834	3,78
ua.if	IF	Ivano-Frankivsk	3,3833	3,38
ua.km	KM	Khmelnytskyi	3,3595	3,36
ua.tp	TP	Ternopil	3,4707	3,47
ua.lv	LV	Lviv	3,5408	3,54
ua.kk	KK	Kharkiv	3,4865	3,49
ua.sm	SM	Sumy	3,2559	3,26
ua.zt	ZT	Zhytomyr	3,261	3,26
ua.rv	RV	Rivne	3,3522	3,35
ua.vo	VO	Volyn	3,3317	3,33

Рисунок 5.18 – Графік, збережений у форматі XLSX

Кожен з зареєстрованих користувачів має свою “роль”, що надає йому певні права і можливості у користуванні сайтом. Користувач “адміністратор” має розширені права. Він має право створювати нові поля у будь-якій з таблиці бази даних. Наприклад, у таблиці Індексу людського розвитку (рисунок 5.19)



Додати human development index model

Region:	<input type="text" value="-----"/>	  
Value:	<input type="text" value="0"/>	
Year:	<input type="text" value="2020"/>	

Зберегти і додати інше

Зберегти і продовжити редагування




ЗБЕРЕГТИ

Рисунок 5.19 – Створення нового запису в таблиці

Також адміністратор має доступ до редагування і видалення будь-яких записів (рисунок 5.20).

Змінити human development index model

ІСТОРІЯ

Region:	<input type="text" value="Чернігівська"/>	  
Value:	<input type="text" value="3.1473"/>	
Year:	<input type="text" value="2004"/>	

Видалити

Зберегти і додати інше

Зберегти і продовжити редагування

ЗБЕРЕГТИ

Рисунок 5.20 – Форма редагування або видалення запису в таблиці

Адміністратор має доступ до перегляда всіх зареєстрованих користувачів (рисунок 5.21), перегляду їхніх персональних даних, редагування, видалення (рисунок 5.22) та зміни їхніх прав у системі (рисунок 5.23).

Виберіть користувач щоб змінити

ДОДАТИ КОРИСТУВАЧ +

Пошук  Пошук

Дія:  Вперед 0 з 2 обрано

<input type="checkbox"/>	ІМ'Я КОРИСТУВАЧА	EMAIL АДРЕСА	ІМ'Я	ПРИЗВИЩЕ	СТАТУС
<input type="checkbox"/>	admin	admin@example.com			✓
<input type="checkbox"/>	amita051197@gmail.com				✗

2 користувачі

#### ВІДФІЛЬТРУВАТИ

За статус персоналу

Всі  
Так  
Ні

За статус суперкористувача

Всі  
Так  
Ні

За активний

Всі  
Так  
Ні

Рисунок 5.21 – Список всіх користувачів

Змінити користувач

Ім'я користувача:

amita051197@gmail.com

Необхідно: 150 або менше символів. тільки букви, цифри та знаки @/./+/\_.

Пароль:

алгоритм: pbkdf2\_sha256 ітерації: 180000 сіль: oxNueo\*\*\*\*\* хеш: v2wBwE\*\*\*\*\*

Raw passwords are not stored, so there is no way to see this user's password, but you can change the password using

#### Персональна інформація

Ім'я:

Прізвище:

Email адреса:

#### Дозволи

Рисунок 5.22 – Форма редагування даних користувача

Дозволи користувача:

В наявності дозволи користувача ?

Фільтр

admin | запис у журналі | Can add log entry

admin | запис у журналі | Can change log entry

admin | запис у журналі | Can delete log entry

admin | запис у журналі | Can view log entry

auth | група | Can add group

auth | група | Can change group

auth | група | Can delete group

auth | група | Can view group

auth | дозвіл | Can add permission

auth | дозвіл | Can change permission

auth | дозвіл | Can delete permission

auth | дозвіл | Can view permission

auth | користувач | Can add user

auth | користувач | Can change user

Обрати всі

Рисунок 5.23 – Список можливих прав доступу користувача

Система реалізована повністю і вирішує всі пославні на початку розробки задачі. Веб-застосунок має простий та лаконічний інтерфейс та дозволяє користувачеві переглядати побудовані діаграми і зберігати їх.

## ВИСНОВКИ

У рамках дипломного проекту було розроблено систему ранжирування територій України за рівнем соціально-екологічних ризиків.

Проведений аналіз існуючих програмних рішень показав, що на момент розробки системи, яка би виконувала ранжування ризиків саме для території України не існує. Це дозволяє стверджувати, що дана робота є актуальною і доцільною.

Програмний модуль, розроблений на базі фреймворку Django, мовою програмування Python, має гнучку архітектуру, тому може бути розширений. До системи можна додавати більше функціональних можливостей, при цьому не змінюючи структуру основного програмного модуля.

Реалізовані наступні сценарії роботи користувача:

- побудова графіків за рівнем показника сталого розвитку за певний період;
- відображення на карті України індексу людського розвитку;
- реєстрація нового користувача в системі;
- вхід існуючого користувача в систему;
- створення нових записів у базі даних адміністратором;
- адміністрування бази користувачів і зміна ролей у системі;

Система може використовуватися науково-дослідницькі інститутами, вищими навчальними закладами тощо.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Програма дій "Порядок денний на XXI століття" ("AGENDA 21"). Ухвалена конференцією ООН з навколишнього середовища і розвитку в Ріо-де-Жанейро (Саміт "Планета Земля" 1992.) – Київ "Інтелсфера": 2000. – 359 с.
2. Формування і розвиток людського потенціалу регіону: сутність, сучасні тенденції та методи оцінювання : монографія / МОН України. Одеський нац. економічний університет /за ред. В.Г. Никифоренка – Одеса: BOI COIU "Атлант", 2018 –103 с.
3. United Nations Development Programme. Human Development Report / New York: Palgrave Macmillan, 2014. — P. 160—163.
4. Статистичний збірник "Регіональний людський розвиток" за 2017 рік. Відповідальний за випуск О. О. Кармазіна. URL: <http://www.ukrstat.gov.ua> (дата звернення: 3.05.2020).
5. Людський розвиток регіонів України: аналіз та прогноз (колективна монографія) / За ред. Е.М. Лібанової. – К. : Ін-т демографії та соціальних досліджень НАН України, 2007. – 367 с.
6. Національна доповідь "Цілі Сталого Розвитку: Україна" URL: [http://un.org.ua/images/SDGs\\_NationalReportUA\\_Web\\_1.pdf](http://un.org.ua/images/SDGs_NationalReportUA_Web_1.pdf) (дата звернення: 3.05.2020).
7. Сен А. Развитие как свобода / пер. с англ. Е. Полецкой; под ред. и с послеслов. Р.М. Нуреева. – Москва : Новое издательство, 2004 – 432 с.
8. Human Development Reports UNDP "Human Development Indices and Indicators: 2018 Statistical Update". URL: <http://hdr.undp.org/en/2018-update> (дата звернення: 04.05.2020)
9. Климко С. Г., Пригода В.М., Сизоненко В.О. Людський капітал: світовий досвід і Україна. К.: Основа, 2006 – 224 с., С.283-284
10. Методика вимірювання регіонального людського розвитку; затверджена Рішенням Президії НАН України та колегії Державної служби статистики

- України від 13.06.2012 р. № 123-м.  
URL: [http://www.idss.org.ua/arhiv/Metodika\\_RLR.zip](http://www.idss.org.ua/arhiv/Metodika_RLR.zip) (дата звернення: 04.05.2020).
11. Статистичний збірник "Регіональний людський розвиток" за 2017 рік / відповідальний за випуск О.О. Кармазіна. URL: <http://www.ukrstat.gov.ua> (дата звернення: 05.05.2020).
  12. Сторінка середовища PyCharm а офіційному сайті компанії JetBrains — Режим доступу: <https://www.jetbrains.com/pycharm/> (дата звернення: 09.05.2020).
  13. Python programming language URL: <https://www.python.org/> (дата звернення: 09.05.2020).
  14. Бейдер Д. Чистый Python. Тонкости программирования для профи. — СПб.: Питер, 2018. — 28 с.
  15. Python in 2018. URL: <https://www.jetbrains.com/research/python-developers-survey-2018>. (дата звернення: 09.05.2020).
  16. Django framework. URL: <https://www.djangoproject.com/start/overview/> . (дата звернення: 10.05.2020).
  17. Fusioncharts. URL: <https://www.fusioncharts.com/> . (дата звернення: 10.05.2020).
  18. Bootstrap. URL: <https://getbootstrap.com/> (дата звернення: 10.05.2020).
  19. Jake Spurlock. Bootstrap / Jake Spurlock. – 2013: O'Reilly Media, 2013. – 128 с.
  20. Marijn Haverbeke. Eloquent JavaScript / Marijn Haverbeke., 2018. – 436 с.
  21. Dr. Axel Rauschmayer. Speaking JavaScript: An In-Depth Guide for Programmers / Dr. Axel Rauschmayer., 2014. – 460 с.
  22. How jQuery Works. URL: <https://learn.jquery.com/about-jquery/how-jquery-works/> (дата звернення: 15.05.2020).
  23. PostgreSQL. URL: <https://www.postgresql.org/about/> (дата звернення: 15.05.2020).
  24. MySQL Products. URL: <https://www.mysql.com/products/> (дата звернення: 15.05.2020).
  25. Regina O. Obe. Up and Running: A Practical Guide to the Advanced Open Source Database / Regina O. Obe, Leo S. Hsu., 2018. – 314 с.

26. Andrew Comeau. MySQL Explained: Your Step-by-step Guide to Database Design / Andrew Comeau., 2015. – 312 с.
27. Microsoft SQL Server 2017. URL: <https://www.microsoft.com/ru-ru/sql-server/sql-server-2017> (дата звернення: 15.05.2020).
28. Хендерсон К. Microsoft SQL Server: структура и реализация / Хендерсон К. – М.: Вильямс, 2005. – 1056 с.
29. The File Information Database. URL: <https://fileinfo.com/> (дата звернення: 15.05.2020).

# ДОДАТОК 1

Система ранжирування територій України за рівнем соціально-екологічних  
ризиків людського розвитку

Специфікація

*УКР.НТУУ «КПІ ім. І.Сікорського».*

Аркушів 5

2020



Позначення	Найменування	Примітки
Документація		
	Записка_Агафонова_TB-361.pdf	Пояснювальна записка
Компоненти		
	human_development/wsgi.py	Модуль для роботи з WSGI
	human_development/urls.py	Модуль для конфігурації URL адрес додатку human_development
	human_development/settings.py	Модуль для конфігурації налаштувань додатку human_development
	accounts/forms.py	Модуль для конфігурації форм додатку accounts
	accounts/urls.py	Модуль для конфігурації URL адрес додатку accounts
	accounts/views.py	Модуль для роботи з представленнями додатку accounts

	map/fusioncharts.py	Модуль для роботи з фреймворком fusioncharts
	map/urls.py	Модуль для конфігурації URL адрес додатку map
	map/views.py	Модуль для роботи з представленнями додатку map
	map/constants.py	Модуль для конфігурації констант додатку map
	map/admin.py	Модуль для роботи з моделями адміністратора в базі для додатку map
	map/models.py	Модуль для роботи з моделями в базі додатку map
	index.html	Представлення, яке відображає інтерактивну карту індексу людського розвитку
	indicator_detail.html	Представлення, яке відображає детальну інформацію про чинник людського розвитку
	indicator_list.html	Представлення, яке

		відображає список чинників людського розвитку
	region_detail.html	Представлення, яке відображає деталі по регіону
	region_list.html	Представлення, яке відображає список всіх регіонів
	sendemail/forms.py	Модуль для конфігурації форм додатку sendemail
	sendemail/views.py	Модуль для роботи з представленнями додатку sendemail
	sendemail/urls.py	Модуль для конфігурації URL адрес додатку sendemail
	base.html	Представлення, яке відображає основу для всіх сторінок
	home.html	Представлення, яке відображає головну сторінку
	navbar.html	Представлення, яке

		відображає навігаційну панель для всіх сторінок
	signup.html	Представлення, яке відображає сторінку з формою реєстрації
	login.htm	Представлення, яке відображає сторінку з формою входу
	password_reset_done.html	Представлення, яке відображає сторінку з успішним відновленням пароля
	password_reset_form.html	Представлення, яке відображає сторінку з формою відновлення пароля

## ДОДАТОК 2

Система ранжирування територій України за рівнем соціально-екологічних  
ризиків людського розвитку

Текст програмного коду

*УКР.НТУУ «КПІ ім. І.Сікорського».*

Аркушів 7

```

from collections import OrderedDict

from django.shortcuts import render
from django.urls import reverse
from django.views import View
from django.views.generic.detail import DetailView
from django.views.generic.list import ListView

from .constants import (HUMAN_DEVELOPMENT_INDEX_MAX,
HUMAN_DEVELOPMENT_INDEX_MIN, INDICATOR_INDEX_MIN,
                        INDICATOR_INDEX_MAX, INDICATORS)
from .fusioncharts import FusionCharts
from .models import IndicatorModel, RegionModel,
HumanDevelopmentIndexModel

# Клас для створення конфігурації інтерактивної карти
class CreateMap:
    color_range_colors = ["#ff0000", "#ff8400", "#FFD74D", "#ccff66", "#11ff00"]
# Конструктор класу
    def __init__(self, caption, subcaption, min_value, max_value, data):
        self.caption = caption
        self.subcaption = subcaption
        self.map_config = OrderedDict()
        self.data_source = OrderedDict()
        self.min_value = min_value
        self.max_value = max_value
        self.data = data
        self.step = (self.max_value - self.min_value) / (len(self.color_range_colors) -
1)
        self.color_range = {"minvalue": str(self.min_value),
                            "code": self.color_range_colors[0],
                            "gradient": "1",
                            "color": []}
        self.generate_data_source()

# Метод для об'явлення основних конфігурацій
    def generate_map_config(self):
        self.map_config["exportEnabled"] = "1"
        self.map_config["caption"] = self.caption
        self.map_config["subcaption"] = self.subcaption
        self.map_config["numbersuffix"] = ""
        self.map_config["includevalueinlabels"] = "1"
        self.map_config["labelsepchar"] = ":"
        self.map_config["theme"] = "fusion"

```

```

# Метод для створення кольорової палітри карти і шкали під нею
def generate_color_range(self):
    for i in range(len(self.color_range_colors) - 1):
        data = {
            "minValue": str(self.min_value + i * self.step),
            "maxValue": str(self.min_value + (i + 1) * self.step),
            "code": self.color_range_colors[i + 1]
        }
        self.color_range["color"].append(data)

# Метод для створення всіх конфігурацій (результатами його роботи
користуємося в подальшому)
def generate_data_source(self):
    self.generate_map_config()
    self.generate_color_range()
    self.data_source["colorrage"] = self.color_range
    self.data_source["chart"] = self.map_config
    self.data_source["data"] = self.data

# Клас для роботи з представленням інтерактивної карти людського розвитку
class MapView(View):

    # Метод для прийняття get запиту від користувача. У ньому прописана логіка
    роботи з інтерактивною картою та взаємодія з базою даних через об'єкти
    моделей RegionModel та HumanDevelopmentIndexModel. На вихід передає дані
    для представлення сторінки 'index.html'
    def get(self, request):
        year = request.GET.get('year', 2017)
        data = []
        for region in RegionModel.objects.all():
            try:
                if region.code == "UA.KC":
                    value =
HumanDevelopmentIndexModel.objects.get(year=year,                                reg
ion_id=RegionModel.objects.get(code="UA.KV").id).value
                elif region.code == "UA.SC":
                    value = HumanDevelopmentIndexModel.objects.get(year=year,
                                                                    region_id=RegionModel.objects.get(cod
e="UA.KR").id).value
                else:
                    value = HumanDevelopmentIndexModel.objects.get(year=year,
region_id=region.id).value
            except HumanDevelopmentIndexModel.DoesNotExist:
                value = None

```

```

        data.append({"id": region.code,
                    "value": value,
                    "tooltext": region.title,
                    "showlabel": True,
                    "link": reverse('region-detail', kwargs={"pk": region.id})})

    generated_map = CreateMap("Індекс Людського Розвитку України",
                              str(year),
                              HUMAN_DEVELOPMENT_INDEX_MIN,
                              HUMAN_DEVELOPMENT_INDEX_MAX,
                              data)
    fusion_map = FusionCharts(type="maps/ukraine",
                              id="ex1",
                              width="880",
                              height="880",
                              renderAt="chart-1",
                              dataFormat="json",
                              dataSource=generated_map.data_source)

    return render(request, 'index.html', {'output': fusion_map.render(),
                                          'chartTitle': 'Індекс Регіонального Розвитку
України'})

```

# Клас для роботи з представленням інтерактивної карти чинника розвитку

```
class IndicatorView(View):
```

# Метод для прийняття get запиту від користувача. У ньому прописана логіка роботи з інтерактивною картою та взаємодія з базою даних через об'єкти моделей RegionModel та IndicatorModel. На вихід передає дані для представлення сторінки 'indicator\_detail.html'

```

    def get(self, request):
        year = request.GET.get('year', 2017)
        title = request.GET.get('title', "Відтворення населення")
        data = []
        for region in RegionModel.objects.all():
            try:
                if region.code == "UA.KC":
                    value = IndicatorModel.objects.get(year=year,
                                                        region_id=RegionModel.objects.get(code="UA.
KV").id,
                                                        title=title
                                                        ).value
                    elif region.code == "UA.SC":

```



```

        value = IndicatorModel.objects.get(year=year,
                                           region_id=RegionModel.objects.get(code="UA.
KR").id,
                                           title=title
                                           ).value
    else:
        value = IndicatorModel.objects.get(year=year, region_id=region.id,
title=title).value
    except IndicatorModel.DoesNotExist:
        value = None
    data.append(
        {"id": region.code,
         "value": value,
         "tooltext": region.title,
         "showlabel": True,
         "link": reverse('region-detail', kwargs={"pk": region.id})})
    generated_map = CreateMap(title,
                              str(year),
                              INDICATOR_INDEX_MIN,
                              INDICATOR_INDEX_MAX,
                              data)
    fusion_map = FusionCharts(type="maps/ukraine",
                              id="ex1",
                              width="880",
                              height="880",
                              renderAt="chart-1",
                              dataFormat="json",
                              dataSource=generated_map.data_source)

    return render(request, 'indicator_detail.html', {'output': fusion_map.render(),
'chartTitle': title})

```

# Клас для роботи зі списком чинників людського розвитку

```

class IndicatorListView(ListView):
    template_name = 'indicator_list.html'
    model = IndicatorModel

```

# Клас для роботи зі списком регіонів

```

class RegionListView(ListView):
    template_name = 'region_list.html'

```

```
model = RegionModel
```

# Метод для створення конфігурації для діаграм за період часу. Повертає словник з даними конфігурацій

```
def generate_time_period_chart(caption, data):
    data_source = OrderedDict()
    chart_config = OrderedDict()
    chart_config["exportEnabled"] = "1"
    chart_config["caption"] = caption
    chart_config["xAxisName"] = "Рік"
    chart_config["yAxisName"] = "Значення"
    chart_config["numberSuffix"] = ""
    chart_config["theme"] = "fusion"
    data_source["data"] = []
    years = []
    for x in data:
        data_source["data"].append({'value': x.value, 'label': str(x.year)})
        years.append(x.year)
    chart_config["subCaption"] = f'[{min(years)}-{max(years)}]'
    data_source["chart"] = chart_config
    return data_source
```

# Клас для роботи з представленням списку регіонів. Відповідає за представлення 'region\_detail.html'

```
class RegionDetailView(DetailView):
    template_name = 'region_detail.html'
    model = RegionModel
```

# Метод для роботи з контекстними даними списку регіонів. Взаємодіє з моделями HumanDevelopmentIndexModel та IndicatorModel. Повертає контекст

```
def get_context_data(self, **kwargs):
    context = super().get_context_data(**kwargs)
    data = HumanDevelopmentIndexModel.objects.filter(region=self.object)
    data_source = generate_time_period_chart(caption=self.object.title + "
область",
                                             data=data)
    fusion_map = FusionCharts("line",
                              "myFirstChart",
                              "800",
                              "400",
                              "myFirstchart-container",
                              "json",
```

```

        data_source)
context['output'] = fusion_map.render()
context['chartTitle'] = 'Індекс Людського Розвитку'

indicator_data = IndicatorModel.objects.filter(region=self.object,
title=INDICATORS[0])

data_source = generate_time_period_chart(caption=self.object.title + "
область",
        data=indicator_data)
fusion_map = FusionCharts("line",
        "mySecondChart",
        "800",
        "400",
        "myFirstchart-container2",
        "json",
        data_source)
context['output2'] = fusion_map.render()
context['chartTitle2'] = INDICATORS[0]
return context

```

## ДОДАТОК 3

Система ранжирування територій України за рівнем соціально-екологічних  
ризиків людського розвитку

Опис програмного модулю

*УКР.НТУУ «КПІ ім. І.Сікорського».*

Аркушів 8

2020

## АНОТАЦІЯ

Основний сервіс системи (map) розроблений з використанням мови Python на фреймворку Django. Сервіс входить до рівня контролера, призначений для обробки запитів користувача та відповідає за обробку отриманих від рівня представлення даних, реалізує всю необхідну логіку веб-застосунку, що відповідає за побудову карт, взаємодіє з рівнем доступу до даних і передає рівню представлення результат обробки.

# ЗМІСТ

Загальні відомості .....	4
Функціональне призначення .....	5
Опис логічної структури.....	6
Вхідні та вихідні дані.....	8

## ЗАГАЛЬНІ ВІДОМОСТІ

Даний модуль представляє собою класи, які надають можливість створювати та завантажувати діаграми.

Для функціонування програми необхідно встановлений Python 3.7, всі залежності описані в Pipfile та система керування базами даних PostgreSQL. В браузері повинен бути увімкнена підтримка JavaScript та cookie файлів.

Мови програмування, які використовувались при розробці: Python та JavaScript.

## **ФУНКЦІОНАЛЬНЕ ПРИЗНАЧЕННЯ**

Розроблений програмний модуль призначений для обміну даними з рівнем доступу до даних та рівнем представлення.

Програмний модуль виконує такі задачі:

- створення та збереження діаграм;
- валідація вхідних даних;
- передача необхідних колекцій у рівень представлення.



# ОПИС ЛОГІЧНОЇ СТРУКТУРИ

Рівень бізнес-логіки містить набір компонентів, які відповідають за обробку отриманих від рівня представлення даних, реалізує всю необхідну логіку додатка, всі обчислення, взаємодіє з рівнем доступу до даних і передає рівню представлення результат обробки.

Рівень бізнес-логіки включає в себе об'єкти для передачі даних (Django ORM) з шару доступу до даних в шар представлення: `IndicatorModel`, `HumanDevelopmentIndexModel`, `RegionModel`.

Рівень бізнес-логіки також включає в себе сервіси, які забезпечують зв'язок між шаром представлення та доступу до даних, валідацію даних та виконують усі основні операції в програмі:

1. `RegionDetailView` – клас для роботи з представленням списку регіонів. Містить наступні поля та методи:
  - a. `get_context_data` – метод для роботи з контекстними даними списку регіонів. Взаємодіє з моделями `HumanDevelopmentIndexModel` та `IndicatorModel`. Повертає контекст;
  - b. `template_name` – поле, що відповідає за представлення класу `'region_detail.html'`;
  - c. `model` – поле, що відповідає за модель класу `RegionModel`;
2. `generate_time_period_chart` – метод для створення конфігурації для діаграм за період часу. Повертає словник з даними конфігурацій відбувається зв'язок з рівнем доступу до даних;
3. `RegionListView` – клас для роботи зі списком регіонів. Містить наступні поля та методи:
  - a. `template_name` – поле, що відповідає за представлення класу `'region_list.html'`;
  - b. `model` – поле, що відповідає за модель класу `RegionModel`;
4. `IndicatorListView` – клас для роботи зі списком регіонів. Містить наступні поля та методи:

- a. `template_name` – поле, що відповідає за представлення класу `'indicator_list.html'`;
  - b. `model` – поле, що відповідає за модель класу `IndicatorModel`;
5. `IndicatorView` – клас для роботи з представленням інтерактивної карти чинника розвитку. Містить наступний метод:
- a. `get` – метод, що приймає запит від користувача. У ньому прописана логіка роботи з інтерактивною картою та взаємодія з базою даних через об'єкти моделей `RegionModel` та `IndicatorModel`. На вихід передає дані для представлення сторінки `'indicator_detail.html'`
6. `MapView` – клас для роботи з представленням інтерактивної карти людського розвитку. Містить наступний метод:
- a. `get` – метод, що приймає запит від користувача. У ньому прописана логіка роботи з інтерактивною картою та взаємодія з базою даних через об'єкти моделей `RegionModel` та `HumanDevelopmentIndexModel`. На вихід передає дані для представлення сторінки `'index.html'`
7. `CreateMap` – клас для створення конфігурації інтерактивної карти. Містить наступні поля та методи:
- a. `__init__` – конструктор класу, приймає назву карти (`str`), підназву карти (`str`), найменше значення(`int`), найбільше значення(`int`), дані карти (`dict`). Містить наступні методи:
  - b. `generate_map_config` – метод для об'явлення основних конфігурацій
  - c. `def generate_color_range` – метод для створення кольорової палітри карти і шкали під нею
  - d. `generate_data_source` – метод для створення всіх конфігурацій даних

## ВХІДНІ ТА ВИХІДНІ ДАНІ

Модуль отримує вхідні дані з шару представлення через параметри запитів. Вихідні дані передаються з шару доступу до даних у шар представлення через значення, які повертають методи сервісу.

Класи сервісу призначені для передачі даних у шар представлення, а саме дані для створення інтерактивних карт або діаграм, що потім передаються у клас FusionCharts, який генерує HTML текст з цих даних.

Наприклад, клас для створення інтерактивної карти України: CreateMap(caption, subcaption, min\_value, max\_value, data) – вхідні дані: назва графіка типу str, підназва графіка типу str, мінімальне значення даних на карті типу float, максимальне значення даних на карті типу float, дані для заповнення карти типу dict; вихідні дані: конфігурація інтерактивної карти типу OrderedDict.

Вихідні дані методу generate\_time\_period\_chart(caption, data) представлені у вигляді OrderedDict та представляють собою конфігурацію для графіків за певний період часу.